Pembuatan Visualisasi Bentuk Ruang dari Gambar

Anthony Wibisono[1], Liliana[2], Kartika Gunadi[3]
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
Email: runner_up1991@hotmail.com[1], lilian@petra.ac.id[2], kgunadi@petra.ac.id[3]

ABSTRAK: Merencanakan struktur ruangan rumah yang diinginkan tidak semudah yang dapat dibayangkan. Untuk mengatasinya, kebanyakan orang mencari referensi melalui struktur rumah yang sudah ada. Akan tetapi masalah belum dapat teratasi oleh karena keterbatasan manusia dalam menentukan struktur suatu ruangan dengan tepat hanya dengan berada di dalam ruang tersebut, sehingga pada akhirnya terlanjur salah dalam perencanaan struktur rumah.
Untuk mengatasi hal ini, dibuatlah sebuah aplikasi yang dapat memberi visualisasi 360° terhadap suatu ruangan sehingga *user* dapat melihat isi ruangan tanpa harus berada dalam ruangan tersebut. Selain itu, user juga dapat membuat denah dengan hanya menandai *north*, *corner* dan *door* yang ada pada tiap ruangan. Untuk membuat visualisasi isi ruangan dilakukan proses *360 Panorama Stitching*. Sedangkan untuk membuat denah rumah, setiap ruangan akan dibentuk terlebih dahulu dengan menggunakan algoritma rekonstruksi ruang kemudian disatukan menggunakan transformasi matriks 2D.
Melalui hasil pengujian, telah didapatkan contoh panorama dari ruangan dalam rumah yang sudah berbentuk 360° dan denah yang dihasilkan hanya berdasarkan data north, corner, dan door pada tiap ruangan. Denah yang dihasilkan dapat membuat user menyadari struktur suatu ruangan sehingga memberikan visualisasi ruangan kepada *user*. Kelemahan dari aplikasi ini adalah user harus menandai *north*, *corner*, dan *door* dengan tepat untuk menghasilkan denah yang sesuai.

Kata Kunci
Panorama 360, visualisasi, rekonstruksi ruang, dan denah

ABSTRACT: *Planning a desired indoor room structure is not as easy as it seems. In order to overcome this problem, usually people collect references from existing indoor floor plans. However, problems may still persist due to human limitations in determining the correct room structure just by being inside the room, so in the end they planned a false indoor room structure.*
In order to overcome this problem, an application is made to give 360° room visualization to the users so that they may see the contents of the room without being inside the room. Moreover, the users can also make a floor plan by their own just by marking the north, the corners, and the doors in every room. To make 360° room visualization, 360° Panorama Stitching process is conducted. Meanwhile to generate a floor plan, every room will be reconstructed first using room reconstruction algorithm and then all of them will be constructed to a complete floor plan using 2D transformation matrix.
According to experiment results, examples of panorama of indoor rooms which already span up to 360° and examples of floor plan which are generated by only using north, corner, and door data have been acquired. By using the panorama, users may imagine the contents of a room without being inside the room. The generated floor plan may help users realize the structure of rooms, hence giving indoor visualization to the users. The weakness of this application is that the users have to mark north point, corners, and doors correctly to produce a correct floor plan.

Keywords
Panorama 360, visualization, room reconstruction, and floor plan

1. PENDAHULUAN
Seringkali seseorang mengalami kesulitan dalam menentukan bentuk suatu ruang oleh karena kurangnya inspirasi dan referensi. Untuk mendapatkan inspirasi dan referensi tersebut, biasanya kebanyakan orang mencoba untuk melihat struktur rumah yang sudah ada. Akan tetapi dengan hanya melihat dan berada di dalam rumah tersebut, masalah belum dapat teratasi oleh karena ketidakmampuan seseorang dalam menentukan kecocokan bentuk, posisi, dan proporsi ruangan terhadap ruangan lainnya, sehingga pada akhirnya terlanjur salah dalam perencanaan struktur rumah, padahal aspek – aspek tersebut sangat penting untuk keberlangsungan kegiatan rumah tangga sehari – hari, baik dalam segi kenyamanan maupun efisiensi.
Salah satu solusi atas permasalahan tersebut adalah visualisasi ruangan, di mana *user* bisa merasakan seakan – akan berada di dalam ruangan tanpa harus berada di dalam ruangan tersebut. Pembuatan visualisasi ruangan dapat dibuat dengan adanya gambar – gambar yang mewakili seluruh isi ruangan. Bentuk suatu ruangan dan proporsisnya terhadap ruangan lainnya dapat dilihat melalui denah secara keseluruhan yang dapat dibentuk dengan memanfaatkan posisi *room corner* dan pintu dalam suatu ruangan yang dapat ditentukan ketika visualisasi sedamg berjalan. Sehingga dengan adanya aplikasi khusus untuk merekonstruksi bentuk dan posisi ruangan – ruangan dalam suatu rumah, permasalahan tersebut dapat diselesaikan.
Oleh karena itu, dibuatlah aplikasi yang dapat merekonstruksi setiap ruangan dalam suatu rumah. User

hanya perlu berada dalam ruangan dan mengambil foto – foto yang dapat mewakili seluruh isi ruangan, lalu aplikasi akan memproses foto – foto tersebut menjadi visualisasi ruangan. Selain itu *user* juga dapat melihat denah rumah setelah semua ruangan telah disatukan. Dengan adanya aplikasi ini, *user* akan sangat terbantu dalam menentukan bentuk, posisi dan proporsi suatu ruangan terhadap ruangan lainnya oleh karena kemudahan dalam melihat seluruh isi rumah, baik tiap ruangan maupun secara keseluruhan meski tidak berada di dalam rumah tersebut.

2. TEORI PENUNJANG

2.1. *360° Panorama Stitching*

Untuk melakukan visualisasi ruangan, langkah pertama yang perlu dilakukan adalah membuat *panorama 360* yang dapat mewakili seisi ruangan. *Panorama* adalah hasil dari penyatuan beberapa gambar yang saling memiliki korelasi menjadi representasi luas suatu pemandangan atau penampakan dalam bentuk 1 kesatuan gambar yang utuh. Terdapat banyak jenis *panorama*, antara lain *linear panorama*, *cylindrical panorama*, *spherical panorama*, dan sebagainya. Untuk membuat visualisasi bentuk ruang, *user* perlu melihat sekeliling ruangan secara 360°, sehingga dibutuhkan jenis *panorama* yang dapat mewakili 360° isi ruangan, yaitu *cylindrical panorama*.

Pembuatan *cylindrical panorama* terlebih dahulu dilakukan dengan memproyeksikan setiap gambar yang akan disatukan kepada bentuk silinder [2]. Metode proyeksi ini disebut dengan *cylindrical projection*. Setiap pixel pada gambar akan dipindahkan menuju koordinat baru, sehingga pada akhirnya setiap gambar akan berbentuk seperti kulit silinder. Secara matematis, berikut ini adalah persamaan yang digunakan untuk melakukan *cylindrical projection:*

$$x' = f\,tan^{-1}\left(\frac{x}{f}\right)$$

$$y' = f\left(\frac{y}{\sqrt{x^2 + f^2}}\right)$$

dimana x' dan y' merupakan koordinat baru, x dan y merupakan koordinat lama, dan f merupakan *focal length* dari kamera yang digunakan.

Ketika semua gambar telah dikonversi menggunakan *cylindrical projection*, yang perlu dilakukan berikutnya adalah menyatukan masing – masing gambar tersebut. Untuk melakukannya, digunakan SIFT sebagai *feature detector* yang dapat mendeteksi *features* atau titik – titik penting pada gambar [3]. *Features* ini kemudian dipakai sebagai acuan dalam melakukan *stitching* antar gambar. Semua *features* pada suatu gambar akan dicocokkan dengan *features* pada gambar lain menggunakan *feature matching* [4]. Lalu hasil dari *feature matching* tersebut digunakan untuk menghasilkan *homography matrix* dengan menggunakan algoritma RANSAC [4]. *Homography matrix* merupakan matriks transformasi untuk mentransformasi suatu gambar agar dapat dicocokkan dengan gambar lain [5]. Akan tetapi oleh karena jenis *panorama* yang digunakan adalah *cylindrical panorama*, hanya elemen translasi saja yang diambil dari *homography matrix* yang dihasilkan. Hal ini dikarenakan rotasi dari kamera adalah translasi dari selimut silinder.

Dengan menjalankan SIFT, *feature matching*, dan RANSAC, *panorama* sudah dapat dihasilkan, namun akan terdapat banyak *obvious seams* yang disebabkan perbedaan cahaya dalam pengambilan gambar. Untuk menghilangkan *obvious seams*, dilakukan *alpha blending* agar setiap gambar dapat terlihat menyatu [6]. Berikut ini adalah persamaan dari *alpha blending*:

$$g(x) = (1 - \alpha)f_0(x) + \alpha f_1(x)$$

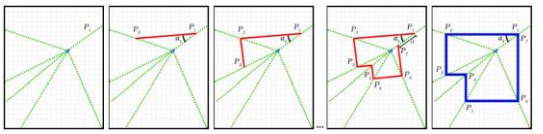
dimana $f_0(x)$ merupakan gambar pertama, $f_1(x)$ merupakan gambar kedua, dan α merupakan *alpha* yang berkisar dari 0 – 1.

Panorama 360 kemudian dapat dihasilkan dengan melakukan langkah terakhir yaitu *cropping*. Agar *panorama* dapat mewakili 360 isi ruangan, gambar pertama akan ditambahkan pada akhir gambar hasil. Kemudian gambar hasil tersebut akan dipotong di kiri dan kanan sebanyak setengah dari lebar gambar pertama. Hal ini menyebabkan akhir dari *panorama* dapat menyatu dengan awalnya ketika *panorama* ditampilkan.

2.2. *Room Reconstruction*

User kemudian dapat menandai posisi *north*, *corners* dan *doors* pada *panorama* yang dihasilkan. Rekonstruksi ruangan dapat dilakukan dengan memanfaatkan posisi *corner* yang sudah di-*input* oleh *user* [1].

Algoritma rekonstruksi ruangan bertugas untuk mencari besar α (derajat dari *corner* pertama menuju *corner* kedua) yang paling sesuai untuk pembentukan ruangan. Nilai α akan dimulai dari 0 dan bertambah sebanyak 0.5 pada setiap iterasi. Penentuan posisi *corner* dilakukan dengan menggunakan perpotongan garis antara derajat *corner* dan arah *corner* sebelumnya. Ketika iterasi selesai, jarak antara *corner* pertama dan terakhir dapat dihitung. Semakin kecil jarak tersebut maka semakin dekat juga rekonstruksi dengan bentuk asli. Iterasi selesai ketika nilai α mencapai 360. Nilai α yang telah didapatkan kemudian akan dipakai untuk merekonstruksi ruangan dalam bentuk gambar.



Algorithm 1 Calculating optimal wall configuration

```
P_0 = Origin
P_1 = Unit distance from P_0 along the first ray
N = Number of corners
minDistance = MAX_FLOAT
for α = 0 → 360 step 0.5 do
    for i = 2 → N do
        θ_i = direction of i^th corner                ▷ from (2)
        P_i ← intersect(P_{i−1}, α + (i − 1) × 90, P_0, θ_i)
    end for
    distance = getDistance(P_1, P_N)
    if distance < minDistance then
        minDistance ← distance
        minAngle ← angle
    end if
end for
Return minAngle
```

Gambar 1. Algoritma Rekonstruksi Ruangan
Sumber: Sankar (2012, p.5)

2.3. *Floor Plan Generator*

Pembuatan denah rumah dapat dilakukan dengan menyatukan masing – masing rekonstruksi ruangan yang telah dibentuk [1]. Sebelum menyatukan setiap ruangan yang ada, masing – masing ruangan akan dirotasi agar tegak lurus terhadap vertikal dan horizontal. Berikutnya, penyatuan rekonstruksi ruangan dilakukan berdasarkan 2 aspek, yaitu posisi dan proporsi ukuran suatu ruangan terhadap ruangan lainnya.

Untuk memenuhi aspek posisi, *user* harus selalu mendefinisikan suatu ruangan apabila ingin berpindah ruangan melalui suatu pintu untuk pertama kalinya. Oleh karena itu, ruangan yang bersebelahan akan selalu saling mengetahui informasi mengenai ke mana ruang yang dituju apabila *user* berniat untuk melewati suatu pintu. Ketika pembuatan denah rumah, pintu di suatu ruangan akan ditranslasi menuju ke pintu ruangan sebelumnya, sehingga secara otomatis akan mentranslasi seisi ruangan. Dengan begitu setiap ruangan akan selalu berdempetan dengan pintu ruang lain.

Sedangkan untuk memenuhi aspek proporsi ruangan, dilakukan perbandingan ukuran pintu pada rekonstrusi ruangan yang bersebelahan. Ukuran pintu yang menjadi patokan adalah ukuran pintu di ruangan yang pertama kali didefinisikan oleh *user*, sehingga ruangan lain hanya perlu menyesuaikan ukuran pintunya dengan ruangan pertama. Ukuran pintu inilah yang akan digunakan sebagai satuan untuk melakukan *scaling* terhadap suatu ruangan.

Menyatukan semua ruangan agar berbentuk denah keseluruhan hanya membutuhkan rotasi, translasi dan *scaling* yang dapat diselesaikan dengan persamaan matriks 2D. Berikut ini adalah persamaan matriks yang digunakan untuk pembentukan denah:

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \begin{pmatrix} S_x * \cos\theta & \sin\theta \\ -\sin\theta & S_y * \cos\theta \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

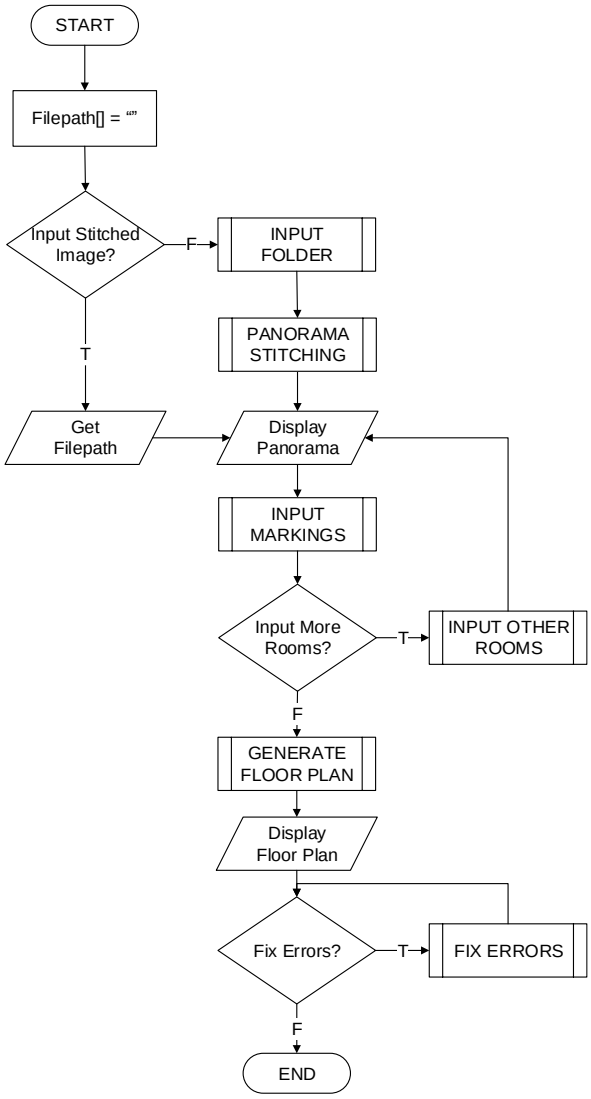
Dimana x_i' dan y_i' adalah koordinat baru hasil transformasi dari x_i dan y_i, S_x dan S_y merupakan faktor skala yang digunakan untuk memperbesar atau memperkecil ruangan serta t_x dan t_y merupakan konstanta yang digunakan untuk mentranslasi ruangan.

3. DESAIN SISTEM

Dalam pembuatan aplikasi terdapat tahap proses yang akan dilakukan seperti diagram alir pada Gambar 2. Proses-proses utama yang akan dilakukan antara lain adalah *input image*, *panorama stitching*, *input markings*, *input rooms*, *generate floor plan*, dan *fix errors*.

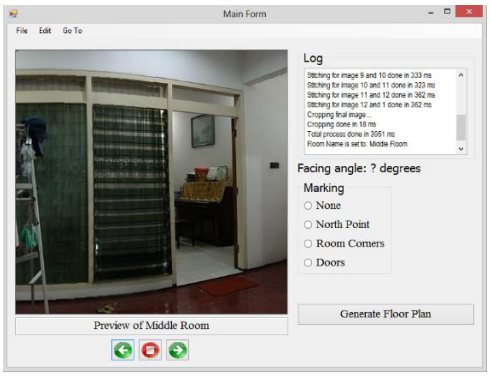
Proses pertama adalah *input image*, dimana *user* dapat memilih untuk meng-input gambar baik berupa *series of images* atau *stitched image*. Apabila *user* memilih *series of images*, proses berikutnya yang akan dijalankan adalah *panorama stitching* untuk menyatukan semua gambar dalam direktori untuk menjadi *panorama*. Sedangkan bila *user* memilih *stitched image*, sistem akan melewati proses *panorama stitching* dan langsung menampilkan *panorama*.

Setelah *panorama* ditampilkan, *user* dapat memberi *north point*, *corner*, dan *door* agar ruangan dapat direkonstruksi dan denah dapat dibentuk. *North point* akan ditandai dengan sebuah arah panah di bagian atas gambar, *corner* akan ditandai dengan sebuah garis vertikal yang dapat digeser oleh *user*, dan *door* akan ditandai dengan 2 garis vertikal yang sejenis dengan garis *corner*. Jika *user* memberi sebuah *door*, sebuah anak panah merah akan muncul yang menandakan bahwa terdapat transisi ruangan yang dapat didefinisikan. Apabila ruangan sudah didefinisikan, arah panah akan berubah warna menjadi hijau. Dengan mengklik anak panah tersebut, *user* dapat berpindah ruangan.

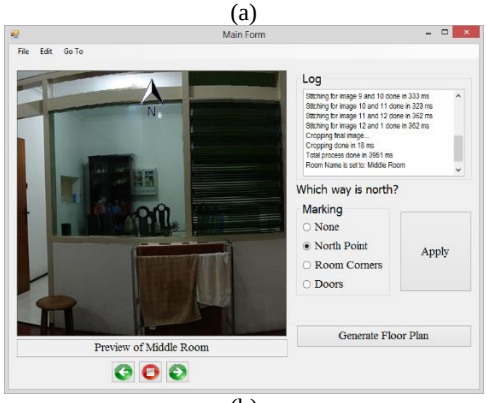


Gambar 2. Diagram Alir Sistem Secara Garis Besar

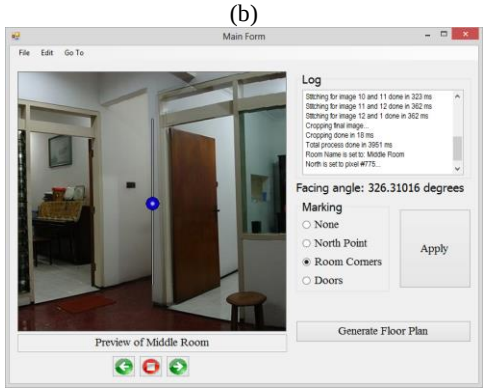
Ketika semua *corner* telah ditandai, *user* dapat memilih *Generate Floor Plan* untuk merekonstruksi semua ruangan dan membuat denah. Sistem akan secara otomatis melakukan rotasi, translasi, dan skala pada setiap rekonstruksi ruangan. Apabila terdapat kesalahan pada denah, *user* dapat mengkoreksi denah tersebut secara manual dengan memilih suatu tembok dan memasangkannya dengan tembok lain.



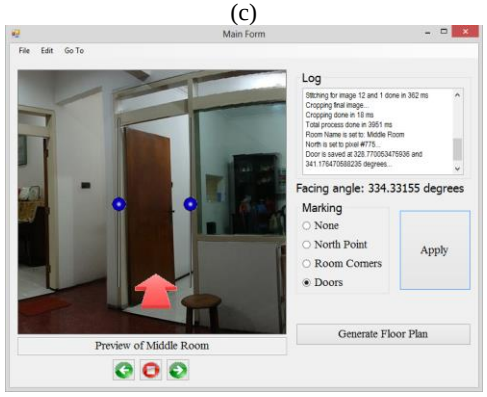
(a)



(b)

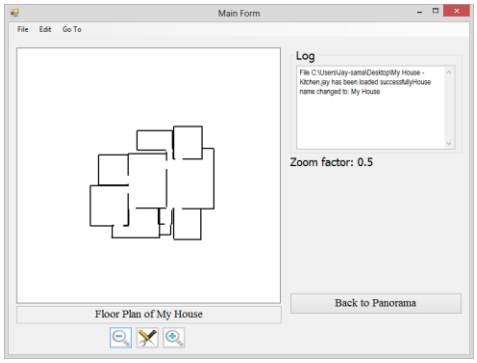


(c)

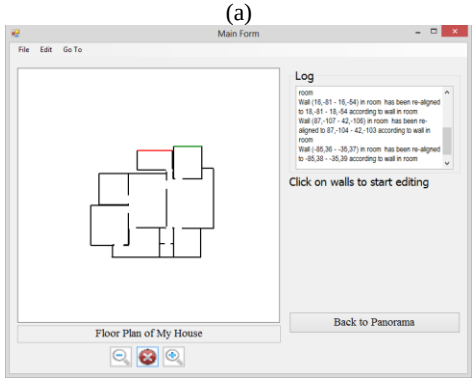


(d)

Gambar 3. *Panorama* dan Pemberian *Marking* (a) Gambar Panorama (b) Tanda *North Point* (c) Tanda *Corner* (d) Tanda *Door* Beserta Anak Panah Penanda Transisi Ruangan



(a)



(b)

Gambar 4. Hasil Proses *Generate Floor Plan* (a) Denah yang Dibentuk Sistem (b) Denah yang Dikoreksi *user* Secara Manual

4. IMPLEMENTASI DAN HASIL

Secara umum sistem telah dapat memenuhi tujuannya yaitu membuat visualisasi bentuk ruang kepada *user*. *Panorama 360* yang dibentuk dapat membuat *user* membayangkan isi ruangan tanpa berada di dalam ruangan tersebut. Terlebih lagi denah yang dihasilkan sudah mirip dengan denah asli, meski terdapat beberapa *error*. Namun *error* tersebut dapat diatasi oleh *user* dengan mengkoreksi denah secara manual. Hasil dari implementasi sistem dapat dilihat pada Gambar 5 dan Gambar 6.

Pada hasil *panorama* yang ditampilkan, terdapat beberapa *ghosting* yang muncul seperti pada uji coba (B) di bagian televisi, (D) di bagian pintu, dan (E) di bagian jendela. Penyebab dari munculnya *ghosting* di antaranya adalah adanya distorsi gambar yang terjadi ketika gambar dikonversi menjadi *cylindrical* dan *keypoints* pada gambar yang tidak mencukupi oleh karena gambar terlalu polos dan tidak berpola yang sering ditemui pada rumah – rumah seperti tembok putih, pintu polos, dan lain – lain.

Hasil uji coba pembentukan denah yang diperlihatkan pada Gambar 6 dilakukan pada 4 macam lingkungan *indoor* seperti rumah dan laboratorium. Dari 4 pengujian tersebut, dapat dilihat bahwa hasil denah yang dibuat sistem pada uji coba (A) dan (B) sudah sesuai dengan denah asli dan hanya memiliki sedikit *error* yang terlihat jelas. Sedangkan pada uji coba (C) dan (D) terdapat cukup banyak *error* berupa bentuk ruang yang terlalu kecil dan berbentuk kurang sesuai dengan denah asli. Terdapat beberapa penyebab yang dapat mengakibatkan kesalahan dalam pembentukan denah, seperti peletakan *north point* yang kurang tepat,

pengambilan posisi gambar yang tidak di tengah ruangan, dan yang merupakan penyebab terbanyak adalah penandaan *corner* dan *door* yang kurang tepat.

Penandaan *corner* dan *door* sangat bergantung pada *panorama* yang dihasilkan. Apabila hasil *panorama* tidak jelas, maka *user* juga akan mengalami kesulitan dalam memberi tanda. Pada uji coba (C) dan (D), terdapat beberapa ruang dengan *corner* atau *door* yang tidak terlihat. Sebagai hasilnya, *user* harus melakukan perkiraan dalam memberi tanda. Penandaan yang salah dapat berakibat pada ukuran dan bentuk ruangan yang salah pula.

(A) (B) (C) (D) (E)

Gambar 5. Hasil *Panorama* (A) Ruang Keluarga (B) Ruang Tidur 1 (C) Ruang Tidur 2 (D) Dapur (E) Ruang Tidur 3

5. KESIMPULAN

Dari pengujian yang sudah dilakukan, dapat ditarik beberapa kesimpulan, antara lain:

- Pembentukan *panorama* sangat tergantung pada isi dan desain dalam suatu ruangan. Ruangan yang hanya berisi sedikit barang dan bermotif polos akan mengalami kekurangan *keypoints* ketika dilakukan proses *panorama stitching*. Bila ini terjadi, hasil *panorama* kemungkinan akan mengalami beberapa kesalahan pada daerah – daerah tersebut.
- Bentuk ruangan yang benar pada proses rekonstruksi ruangan bisa didapatkan ketika *user* dapat menandai semua *corner* dan posisi *north point* dengan tepat. Ruangan – ruangan dengan *corner* yang tidak terlihat, baik karena bentuk ruangan tidak beraturan, tertutup oleh benda lain, maupun pencahayaan yang kurang bagus memiliki resiko mengalami kesalahan ketika rekonstruksi ruangan.
- Peran pintu dalam pembentukan denah keseluruhan sangat penting. Hal ini disebabkan konstanta *scaling* yang digunakan untuk mendapatkan proporsi suatu ruangan dibandingkan ruangan lainnya didapatkan dengan melakukan perbandingan ukuran pintu antar ruang. Kesalahan ukuran atau letak pintu di suatu ruangan akan berakibat pada ukuran ruangan lain.
- Banyak faktor yang dapat menyebabkan kesalahan dalam pembuatan denah, di antaranya adalah kesalahan peletakan *corner* atau *door* baik karena *ghosting* atau tidak terlihat, posisi pengambilan gambar yang tidak di tengah ruangan, dan terdapat posisi *north point* yang kurang tepat. Namun secara umum, masalah yang paling sering dihadapi ketika terjadi kesalahan pembentukan denah adalah peletakan *corner* atau *door* yang tidak sesuai dengan kondisi asli.

6. REFERENCES

[1] Sankar, A. & M. Seitz, S. (2012, October). *Capturing Indoor Scenes with Smartphones*. Paper presented at UIST '12 Proceedings of the 25th annual ACM symposium on User interface software and technology, New York, USA.

[2] Szeliski, R. & Shum, H. Y. (1997, August). *Creating Full View Panoramic Image Mosaics and Environment Maps*. Paper presented at Computer Graphics (SIGGRAPH'97 Proceedings), New York, USA.
[3] Lowe, D. G. (2004, November). *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60, pp. 91-110.
[4] Brown, M. & Lowe, D. G. (2007, August). *Automatic Panoramic Image Stitching using Invariant Features*. International Journal of Computer Vision, vol. 74, pp. 59-73.
[5] Elan, D. (2009, March). *Homography Estimation*. Vancouver: The University of British Columbia.
[6] Burt, P. J. & Adelson, E. H. (1983, October). *A Multiresolution Spline with Application to Image Mosaics*. ACM Transactions on Graphics (TOG), vol. 2, pp. 217-236.

Gambar 6. Hasil Pembentukan Denah (A) Rumah Filbert (B) Laboratorium Pemrograman dan Sistem Informasi (C) Rumah Meliana (D) Rumah Anthony