DAFTAR PUSTAKA

Abrahams JJ, Glassberg RM. (1996). *Dental disease: Frequently unrecognized cause of maxillary sinus abnormalities*. AJR; 166: 1219-1223

Action On Smoking and Health Research Report. (2012). *Tobacco and oral health*. http://ash.org.uk . Diakses pada 29 Maret 2014

Adam Inc. (2014). *Tooth Anatomy*. U.S National Library of Medicine. MedlinePlus. https://www.nlm.nih.gov/medlineplus/ency/imagepages/1121.htm. Diakses pada 11 Januari 2015

Adobe System Inc. (2014). *Reaching and building loyalty with your most valuable mobile customers*. Adobe 2014 Mobile Consumer Survey Results. Adobe System Inc.

Amadeo, Ron. (2012). *A History of Pre-Cupcake Android Codenames*. Android Police. http://www.androidpolice.com/2012/09/17/a-history-of-pre-cupcake-android-codenames/. Diakses pada 4 Juni 2014

Android. (2007). *The Android Story*. Android. https://www.android.com/history/. Diakses pada 4 Juni 2014

Android Developers. (2007). *SDK Archieves*. Android Developers. http://developer.android.com/sdk/older_releases.html. Diakses pada 4 Juni 2014

Android Developers. (2007). *Android, the world's most popular mobile platform*. Android Developers. https://developer.android.com/about/android.html. Diakses pada 4 Juni 2014

Android Developers. (2007). *Codelines, Branches, and Releases*. Android. Developers. http://source.android.com/source/build-numbers.html. Diakses pada 4 Juni 2014

Android Developers. (2007). *The Android Source Code*. http://source.android.com/source/index.html. Android Developers. Diakses pada 4 Juni 2014

Ardhianto Eka, Hadikurniawati Wiwien, Winarno Edy. (2012). Augmented Reality Objek 3 Dimensi dengan Perangkat Artoolkit dan Blender. Jurnal Teknologi Informasi DINAMIK Jul; 17(2): 107-117

Arvanitis T. N., Petrou A., Knight J. F., Savas S., Sotiriou S., Gargalakos M., Gialouri E.. (2009). *Human factors and qualitative pedagogical evaluation of a mobile augmented reality system for science education used by learners with physical disabilities*. Personal and ubiquitous computing; 13(3): 243–250

Azuma R. T. (1997). *A survey of augmented reality*. Presence: Teleoperators and Virtual Environments Aug; 6(4): 355-385

Azuma R.T., Hoff B.R., Howard E. Neely I., Sarfaty R., Daily M. J., Bishop G., Welch L. Vicci, G., Neumann U., You S., Nichols R., Cannon J. (1998). *Making augmented reality work outdoors requires hybrid tracking*. IWAR; pp. 219-224

Azuma R., Neely I., H., Daily M., Leonard J.. (2006). *Performance analysis of an outdoor augmented reality tracking system that relies upon a few mobile beacons*. IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR); pp. 101–104

Badan Penelitian dan Pengembangan Kesehatan. (2008). Hasil Riset Kesehatan Dasar 2007. Jakarta: Departemen Kesehatan Republik Indonesia

Baillot Y., Brown D., Julier S.. (2001). *Authoring of physical models using mobile computers*. Fifth International Symposium on IEEE, Wearable Computers; pp. 39–46

Bajura M., Fuchs H., Ohbuchi R., (1992). *Merging virtual objects with the real world: Seeing ultrasound imagery within the patient*. ACM SIGGRAPH Computer Graphics; 26(2): 203–210

Bell, Donald. (2003). *UML Basics Part II: The Activity Diagram*. IBM Global Services, Rational Software

Benarjee A, Watson T. F.. (2012). Pickard Manual Konservasi Restoratif, Ed. 9. Jakarta: Buku Kedokteran EGC

Bimber O., Raskar R. (2005). *Spatial Augmented Reality: Marging Real and Virtual Words*. A K Peters Ltd

Blender Foundation. (2015). *Blender's Features*. Blender. https://www.blender.org/features/. Diakses pada 26 Juni 2015

Bowden, George. (2000). *Microbial Ecology in Health and Disease: The Microbial Ecology of Dental Caries*. Department of Oral Biology, Faculty of Dentistry. Canada: Taylor & Francis; 12: 138–148

Bower Matt, Howe Cathie, McCredie Nerida, Robinson Austin, Grover David. (2014). *Augmented Reality in education – cases, places and potentials*. Educational Media International; 51(1): 1-15

Blum T., Stauder R., Euler E., Navab N.. (2012). *Superman-like x-ray vision: Towards brain-computer interfaces for medical augmented reality*. IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp.271–272

Breeze, Mez. (2012). *The designer behind the logo*. *TheNextWeb* (TNW). http://thenextweb.com/dd/2012/09/08/the-stories-inspiration-behind-world-famous-logos/. Diakses pada 4 Juni 2014

Business Wire. (2010). *Total Immersion to Unveil First Commercial Markerless Tracking on Mobile at International CTIA WIRELESS 2010® in Las Vegas: CEO Bruno Uzzan Joins Panel on Mobile Augmented Reality, Examines Opportunities to Monetize Mobile*. http://www.businesswire.com/news/home/20100323005910/en/Total-Immersion-Unveil-Commercial-Markerless-Tracking-Mobile#.VYrUbkad8Xg. Diakses pada 1 Mei 2014

Carmigniani J., Furht B., Anisetti M., Ceravolo P., Damiani E., Ivkovic M.. (2011). *Augmented reality technologies, systems and applications*. Multimedia Tools and Applications; 51(1): 341–377

Caudell T. P., Mizell D. W.. (1992). *Augmented reality: An application of heads-up display technology to manual manufacturing processes*. Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences; 2: 659–669

Cavallaro R.. (1997). *The foxtrax hockey puck tracking system*. IEEE Computer Graphics and Applications; 17(2): 6–12

Cavallaro R., Hybinette M., White M., Balch T.. (2011). *Augmenting live broadcast sports with 3d tracking information*. IEEE MultiMedia; pp. 38–47

Clark A., Dunser A.. (2012). *An interactive augmented reality coloring book*. IEEE Symposium on 3D User Interfaces (3DUI); pp. 7–10

Daily M., Cho Y., Martin K., Payton D.. (2003). *World embedded interfaces for human-robot interaction*. Proceedings of the 36th Annual Hawaii International Conference on System Sciences, IEEE; pp. 6–pp

Dantone M., Bossard L., Quack T., Van Gool L. (2011). *Augmented Faces*. IEEE International Workshop on Mobile Vision (ICCV 2011)

DiMarzio, Jerome. (2008). *Android: A Programmer's Guide.* New York City, New York: The McGraw-Hill Companies

Ecma International. (2006). *C# Language Specification* 4th ed. Geneva: Ecma International

Education for You. (2014). Use Case, Activity Diagram dan Sequence Diagram. http://eduedu93.blogspot.com/2014/02/use-case-activity-diagram-dan-sequence.html. Diakses pada 11 January 2015

Elvezio, Carmine. (2014). *Installing Unity and Vuforia.* Computer Graphics and User Interfaces (CGUI) Lab, Columbia University. http://graphics.cs.columbia.edu/wiki/doku.php?id=wiki:installing_unity_and_vuforia. Diakses pada 12 Juni 2014

Euromonitor International (2015). *Key Traits of Android and Apple Users*. Passport, Euromonitor International.

Fear, Ed. (2009). *How three guys found Unity in starting a revolution*. Develop. http://www.develop-online.net/analysis/united-they-stand/0116643. Diakses pada 12 Juni 2014

Felker Donn, Dobbs Joshua. (2011). *Android Application Development for Dummies.* Indianapolis, Indiana: Wiley Publishing, Inc.

Fiala. M. (2005). *Artag, a fiducial marker system using digital techniques*. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition.

Fjeld M., Voegtli B. M.. (2002). *Augmented chemistry: An interactive educational workbench*. Proceedings. International Symposium on Mixed and Augmented Reality (ISMAR), IEEE; pp. 259–321

Foxlin E., Altshuler Y., Naimark L., Harrington M.. (2004). *Flighttracker: a novel optical/inertial tracker for cockpit enhanced vision*. in Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society; pp. 212–221

Gargenta, Marko. (2011). *Learning Android.* Gravenstein Highway North, Sebastopol, CA: O'Reilly Media

Green S. A., Billinghurst M., Chen X., Chase G. (2008). *Human-robot collaboration: A literature review and augmented reality approach in design*. International Journal of Advanced Robotic Systems; 5(1)

Gumster, Jason. (2009). *Blender for Dummies*. Indianapolis, Indiana: Wiley Publishing, Inc

Günthner, W. A., Blomeyer, N., Reif, R., Schedlbauer, M. (2009). *Pick-by-Vision: Augmented Reality unterstützte Kommissionierung.* Germany: fml-Lehrstuhl für Fördertechnik Materialfluss Logistik

Haas, John K.. (2014). *A History of the Unity Game Engine*. Worcester, Massachusetts: Worcester Polytechnic Institute

Hackborn, Dianne. (2012). *Android Police Google+: A History of Pre-Cupcake Android Codenames*. Google+. https://plus.google.com/+AndroidPolice/posts/MwDsoxDJ9qQ. Diakses pada 4 Juni 2014

Haller M., Billinghurst M., Thomas B. (2007). *Emerging technologies of augmented reality: Interfaces and design*. Hershey, PA: Idea Group Publishing

Ibañez Alexandro S.,Figueras Josep P.. (2013). *Vuforia v1.5 SDK: Analysis and evaluation of capabilities*. Barcelona, Spain: Polytechnic University of Catalonia.

IBM. (1969). *Data Processing Techniques*. White Plains, New York: International Business Machine Corporation

Jacko, Julie A., Andrew Sears (2007). *Human-Computer Interaction Fundamentals*. CRC Press.

Johnson N. W., Bain C. A.. (2000). *Tobacco and oral disease. EU-Working Group on Tobacco and Oral Health*. Brit Dent J; 189: 200-206.

Johnson S. K., Huizenga G., Pulavarty B.. (2005). *Performance Tuning for Linux Servers*. Indianapolis, Indiana: IBM Press

Julier S., Lanzagorta M., Baillot Y., Rosenblum L., Feiner S., Hollerer T., Sestito S.. (2000). *Information filtering for mobile augmented reality*. IEEE and ACM International Symposium on Augmented Reality (ISAR); pp. 3–11

Jun Yeon Ma, Jong Soo Choi. (2007). *The Virtuality and Reality of Augmented Reality*. Journal of Multimedia Feb; 2(1)

Kim W. S.. (1995). *Virtual reality calibration and preview/predictive displays for telerobotics*. Pasadena: California Insitute of Technology

Kincade, K. (2014). *Cityscape: An Augmented Reality Application for Displaying 3D Maps on a Mobile Device*. MS Project report, Department of Electrical Engineering and Computer Science, Colorado School of Mines.

Kokten G, Balcioglu H, Buyukertan M. (2003). *Supernumerary fourth and fifth molars: a report of two cases*. J Contemp Dent Pract Nov 15; 4(4): 67-76

Krevelen D. Van, Poelman R.. (2010). *A survey of augmented reality technologies, applications and limitations*. International Journal of Virtual Reality; 9(2): 1

Krishnan V. (2012). *Systemic Affliction of Oral Focal Sepsis*. J Indian Aca Oral Med Radiol; 24(2): 137-141

Kroenke, David M.. (2005). *Database Processing*. Jakarta:Erlangga. 1(9):60.

Lazuardy, Senja. (2012). Augmented Reality: Masa Depan Interaktivitas. http://tekno.kompas.com/read/2012/04/09/12354384/augmented.reality.masa.depan.interaktivitas. Diakses pada 1 Mei 2014

Layar. (2009). 2D and 3D objects. https://www.layar.com/documentation/browser/howtos/2d-and-3d-objects/. Diakses pada 1 Mei 2014

Liang, William W.-Y. (2010). *System Integration for the Android Operating System*. Departement Computer Science and Information of Engineering. Taipei, Taiwan: National Taipei University of Technology.

Livingston M. A., Rosenblum L. J., Julier S. J., Brown D., Baillot Y., Swan II J. E., Gabbard J. L., Hix D. (2002). *An augmented reality system for military operations in urban terrain*. Proceedings of Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) Dec 2-5; (1): 89

Lux J, Lavigne S. (2004). *Your mouth – portal to your body: CDHA position paper on the links between oral health and general health. Part I.* Probe; 38(3): 115-34

Lux Judy. (2006). *Review of the Oral Disease-Systemic Disease Link. Part 1: Heart Disease, Diabetes*. Canadian Journal of Dental Hygiene. November-December, 40(6):288-342

Lux Judy. (2007). *Review of the Oral Disease-Systemic Disease Link. Part II: Preterm Low Birth Weight Babies, Respiratory Disease*. January-February; 41(1):8-21

Matthew, Debora C. (2002). *The Relationship Between Diabetes and Periodontal Disease*. Journal of the Canadian Dental Association; 68(3): 161-4

Marsh PD, Martin M. Oral Microbiology. 5th ed. (2009). Churchill Livingstone Elsevier, London, UK; 12

Meurman Jukka H, Sanz Mariano, Janket Sok-Ja. (2004). *Oral Health, Atherosclerosis, and Cardiovascular Disease*. Crit Rev Oral Biol Med; 15(6):403-413

Milgram, P., F. Kishino . (1994). *A Taxonomy of Mixed Reality Visual Displays*. IEICE Trans. Information Systems Dec; E77-D (12): 1321–1329

Morrill, Dan. (2013). *Dan Morrill Google+: The first Android mascot logos*. Google+. https://plus.google.com/+DanMorrill/posts/WrBh6fsChfH. Diakses pada 4 Juni 2014

Dr. Che Noor Aini bt Che Omar, Dr. Laila bt Abd Jalil. (2012). *Gigi Kekal Tumbuh*. MyHealth Kementerian Kesihatan Malaysia. http://www.myhealth.gov.my/index.php/gigi-kekal-lewat-tumbuh. Diakses pada 11 January 2015

NASA. (2013). *Virtual Reality Laboratory*. NASA. http://www.nasa.gov/centers/johnson/engineering/robotics_simulation/virtual_reality/index.html Diakses pada 1 Mei 2014

Nelson, Fritz., Yam, Marcus. (2014). *A Short History of Augmented Reality:* The Past, Present, And Future Of VR And AR: The Pioneers Speak:. http://www.tomshardware.com/reviews/ar-vr-technology-discussion,3811-3.html. Diakses pada 1 Mei 2014

Nelson Stanley J., Ash Jr. Major M.. (2010). *Wheeler's dental anatomy, physiology, and occlusion*. St. Louis, Missouri: Saunders Elsevier

Nunn J., Morris J., Pine C., Pitts N. B., Bradnock G., Steele J.. (2000). *Adult Dental Health Survey: The condition of teeth in the UK in 1998 and implications for the future*. British Dental Journal Dec 23; 189(12): 639–644

Nvidia Corporation. (2014). *GPU-Accelerated Augmented Reality*. Nvidia. http://www.nvidia.com/content/cuda/spotlights/nestor-gomez-artefacto.html. Diakses pada 8 Juli 2015

Object Management Group (OMG). (2007). *Unified Modeling Language: Superstructure version 2.1.1*. Needham, Massachusetts: Object Management Group

Oktari Permata Sari Ni Komang, Crisnapati Padma Nyoman, Antara Kesiman Made Windu, Gede Sunarya I Made. (2014). Pengembangan Aplikasi Augmented Reality Book Penegenalan Tata Letak Bangunan Pura Goa Lawah dan Pura Goa Gaja. JPTK, UNDIKSHA Jul; 11(2): 75–86

Papagiannakis G., Singh G., Magnenat-Thalmann N..(2008). *A survey of mobile and wireless technologies for augmented reality systems*. Computer Animation and Virtual Worlds; 19(1): 3–22

Petersen PE, Lennon MA. (2003). *Continuous improvement of oral health in the 21st century – the approach of the WHO Global Oral Health Programme World Health Organization.* The World Oral Health Report 2003; 32: 4-9.

Petersen PE, Lennon MA. (2004). *Effective use of fluorides for the prevention of dental caries in the 21st century: the WHO approach.* Community Dentistry and Oral Epidemiology; 32: 139-321

Petersen, K., Wohlin, C., Baca, D. (2009). "The Waterfall Model in Large-Scale Development", *Proceedings of 10th International Conference, PROFES 2009, pp. 386-400*.

Permensesneg (2011). Peraturan Menteri Sekretaris Negara Nomor 2 Tahun 2011: Organisasi dan Tata Kerja Kementerian Sekretariat Negara. http://www.setneg.go.id. Diakses pada 22 Agustus 2015

Permensesneg (2011). Peraturan Menteri Sekretaris Negara Nomor 21 tahun 2011 tentang Petunjuk Pelaksanaan Penyusunan dan Evaluasi *Standard Operating Procedures* Kementerian Sekretariat Negara. http://www.setneg.go.id. Diakses pada 22 Agustus 2015

Permensesneg (2015). Peraturan Menteri Sekretaris Negara RI Nomor 3 Tahun 2015: Organisasi dan Tata Kerja Kementerian Sekretariat Negara. http://www.setneg.go.id. Diakses pada 22 Agustus 2015

Pressman, R.S. (2010). *Software Engineering: a Practioner's Approach 7th Edition*, McGraw-Hill Higher Education.

Prestonwood Dental. (2011). *Tooth Decay (A Preventable Disease)*. http://prestonwooddental.com/dental-guide/tooth-decay-a-preventable-disease/. Diakses pada 12 Juni 2014

Qualcomm Inc. (2010). *Qualcomm Extends Augmented Reality Platform with Unity Technologies*. Qualcomm Press Releases. https://www.qualcomm.com/news/releases/2010/11/10/qualcomm-extends-augmented-reality-platform-unity-technologies. Diakses pada 12 Juni 2014

Qualcomm Inc. (2010). *Qualcomm Introduces Mobile Augmented Reality Platform and Software Development Kit*. Qualcomm Press Releases. https://www.qualcomm.com/news/releases/2010/06/30/qualcomm-introduces-mobile-augmented-reality-platform-and-software. Diakses pada 12 Juni 2014

Qualcomm Inc. (2010). *Vuforia Augmented Reality SDK*. Qualcomm Developer Network. https://developer.qualcomm.com/software/vuforia-augmented-reality-sdk. Diakses pada 12 Mei 2014

Qualcomm Inc. (2011). *Developing with Vuforia*. Qualcomm Vuforia Developer Portal. http://developer.vuforia.com/. Diakses pada 12 Mei 2014

Qualcomm Inc. (2011). *Developing with Vuforia*. Qualcomm Vuforia Developer Portal. https://developer.vuforia.com/library//articles/Training/Image-Target-Guide. Diakses pada 12 Mei 2014

Qualcomm Inc. (2011). *Vuforia SDK Native - Android & iOS API Reference*. Qualcomm Vuforia Developer Portal. https://developer.vuforia.com/resources/api/main. Diakses pada 12 Mei 2014

Ramadhan, Ardhya G., (2010). Serba Serbi Kesehatan Gigi dan Mulut. Jakarta: Bukune

Ramesh Raskar, Greg Welch, Henry Fuchs. (1998). *Spatially Augmented Reality*. San Francisco: First International Workshop on Augmented Reality.

Reibel J. (2003). *Tobacco and oral diseases: an update on the evidence, with recommendations*. Med Princ Pract; 12 (sppl 1):22-32

Reinhart G., Patron C.. (2003). *Integrating augmented reality in the assembly domain - fundamentals, benefits and applications*. (CIRP) Annals - Manufacturing Technology; 52(1): 5 – 8

Riccitiello, John (2014). *John Riccitiello sets out to identify the engine of growth for Unity Technologies (interview)*. VentureBeat: Interview with Dean Takahashi. http://venturebeat.com/2014/10/23/john-riccitiello-sets-out-to-identify-the-engine-of-growth-for-unity-technologies-interview/. Diakses pada 12 Mei 2014

Roberts D. W., Strohbehn J. W., Hatch J. F., Murray W., Kettenberger H..(1986). *A frameless stereotaxic integration of computerized tomographic imaging and the operating microscope.* Journal of neurosurgery; 65(4) pp. 545–549

Roeslan BO. (2002). Imunologi karies. In : Imunologi oral Kelainan di dalam rongga mulut. Jakarta: Balai Penerbit FK UI; 139-41

Rolland Jannick, Rich Holloway, and Henry Fuchs. (1994). *A Comparison of Optical and Video See-Through Head-Mounted Displays*. SPIE Proceedings: Telemanipulator and Telepresence Technologies (Boston, MA, 31 October - 4 November); 2351: 293-307

Ross Michael, Pawlina Wojciech. (2011). *Histology: a text and atlas with correlated cell and molecular biology*. Philadhelpia, US: Wolters Kluwere Health, Lippincott Williams & Wilkins

Royce, W. W. (1970). "Managing The Development of Large Software Systems", *Proceedings of IEEE WESCON, pp. 1-9*

Sanders B., Crowe R., Garcia E.. (2004). *Defense advanced research projects agency – smart materials and structures demonstration program overview*. Journal of Intelligent Material Systems and Structures; 15(4): 227–233

Sawers, Paul. (2011). *Augmented Reality: The Past, Present, And Future*. http://thenextweb.com/insider/2011/07/03/augmented-reality-the-past-present-and-future/. Diakses pada 1 Mei 2014

Setyati Endang, Alexandre David, Widjaja Daniel. (2012). *Face Tracking Implementation with Pose Estimation Algorithm in Augmented Reality Technology*. Procedia - Social and Behavioral Sciences; (57): 215 – 222

Sielhorst T., Feuerstein M., Navab N.. (2008). *Advanced medical displays: A literature review of augmented reality*. Journal of Display Technology; 4(4): 451–467

Silva R., J. Oliveira C., Giraldi G. A.. (2003). *Introduction to Augmented Reality*. Brazil: National Laboratory for Scientific Computation

Sima Corneliu, Glogauer Michael. (2012). *Periodontitis in Patients with Diabetes—A Complication that Impacts on Metabolic Control*. Canada: University of Toronto

Sommerville, I. (2011). *Software Engineering 9th Edition*. Addison-Wesley

Spies R., Ablaßmeier M., Bubb H., Hamberger W.. (2009). *Augmented interaction and visualization in the automotive domain*. Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction, ser. Lecture Notes in Computer Science, J. Jacko, Ed. Springer Berlin Heidelberg; 5612: 211–220

Stephan RM, Miller BF. (1943). *A quantitative method for evaluating physical and chemical agents which modify production of acids in bacterial plaques on human teeth*. J Dent Res; 22: 45-51

Sterneckert, Alan B. (2003) *Critical Incident Management*. Boca Raton, Florida, US: Auerbach Publications

Stilman M., Michel P., Chestnutt J., Nishiwaki K., Kagami S., and Kuffner J.. (2005). *Augmented reality for robot development and experimentation*. Robotics Institute. Pittsburg: Carnegie Mellon University

Srigupta, A.A,. (2004). *Perawatan Gigi dan Mulut*. Jakarta: Prestasi Pustaka

Sutherland I. E.. (1968). *A head-mounted three dimensional display*. fall joint computer conference, part I. ACM Dec 9-11; pp. 757–764

Suzuki N., Hattori A., Hashizume M.. (2008). *Benefits of augmented reality function for laparoscopic and endoscopic surgical robot systems navigation.* 1: 6

Tickle M, Milsom K, King D, Kearney-Mitchell P, Blinkhorn A. (2002). *The fate of the carious primary teeth of children who regularly attend the general dental service.* British Dental Journal Feb 23;192(4):219-223

Total Immersion. (2010). *Augmented Reality Technologies*. Total Immerson. http://www.t-immersion.com/augmented-reality/augmented-reality-technologies. Diakses pada 1 Mei 2014

Total Immersion. (2010). *Datasheet D'Fusion Mobile*. Total Immersion.

The Wall Street Journal. (2010). *The Winners, Category by Category: From Computing Systems to Wireless, the Most Innovative Technologies*. http://www.wsj.com/articles/SB10001424052748703904304575497473735761294. The Wall Street Journal. Diakses pada 12 Juni 2014

United States Department of Health and Human Services. (2000). *Oral Health in America: A Report of the Surgeon General.* Rockville, MD: U.S. Department of Health and Human Services, National Institute of Dental and Craniofacial Research, National Institute of Health

Unity Technologies. (2005). *Unity Manual: Learning The Interface*. Unity. http://docs.unity3d.com/Manual/LearningtheInterface.html. Diakses pada 12 Juni 2014

Unity Technologies. (2005). *Unity Pro and Unity Personal Software License Agreement*. Unity. http://unity3d.com/legal/eula. Diakses pada 12 Juni 2014

Unity Technologies. (2005). *Unity Web Player*. Unity. http://unity3d.com/webplayer. Diakses pada 12 Juni 2014

Unity Technologies. (2005). *What's Unity?*. Unity. http://unity3d.com/public-relations. Diakses pada 12 Juni 2014

Urban E. C.. *The information warrior.* (1999). Technology and society. Prentice-Hall, Inc.; pp. 493–501

Vacchetti, L.. Lepetit V., Fua P.. (2004). *Stable Real-Time 3D Tracking using Online and Offline Information*. IEEE Trans. Pattern Anal. Mach. Intell. 26(10): 1385-1391

Vondrak M., Sigal L, Jenkins OC.. (2013).. *Dynamical simulation priors for human motion* tracking. IEEE Trans Pattern Anal Mach Intell, Jan; 35(1): 52-65

Welch Greg, Foxlin Eric. (2002). *Motion Tracking: No Silver Bullet, but a Respectable Arsenal*. IEEE Computer Graphics and Applications; 2(6): 72-171

Wen R., Tay W.-L., Nguyen B. P., Chng C.-B., Chui C.-K..(2014). *Hand gesture guided robot-assisted surgery based on a direct augmented reality interface.* Computer Methods and Programs in Biomedicine; 2(6): 68-80

Wikipedia. (2014). *Flowchart: A simple flowchart representing a process for dealing with a non-functioning lamp.* https://en.wikipedia.org/wiki/Flowchart. Diakses pada Diakses pada 22 Desember 2014

World Health Organization. (1994). *Fluorides and Oral Health.* WHO Technical Report Series No. 846. Geneva: World Health Organization

World Health Organization. (1997). *Oral Health Surveys - Basic Methods.* 4th ed. Geneva: World Health Organization

World Health Organization (2003), *Dental Diseases and Oral Health*. http://www.who.int/mediacentre/factsheets/fs318/en/. Diakses pada 29 Maret 2014

World Health Organization. (2003). *Diet, Nutrition and the Prevention of Chronic Diseases.* WHO Technical Report Series 916. Geneva: World Health Organization.

World Health Organization. (2004). *Challenges to improvement of oral health in the 21st century – the approach of the WHO Global Oral Health Programme.* International Dental Journal; 54: 329–343. Geneva: World Health Organization.

World Health Organization (2012), *Oral Health*. http://www.who.int/mediacentre/factsheets/fs318/en/. Diakses pada 29 Maret 2014

Worn H., Aschke M., Kahrs L.. (2005). *New augmented reality and robotic based methods for head-surgery*. The International Journal of Medical Robotics and Computer Assisted Surgery; 1(3): 49–56

Wu, T. (2000). *Periodontal disease and risk of cerebrovascular disease*. Arc Intern Med; 160: 2749-2755

Yu D., Jin J. S., Luo S., Lai W., Huang Q.. (2010). *A useful visualization technique: A literature review for augmented reality and its application, limitation & future direction*. Visual Information Communication, Springer; pp. 311–337

DAFTAR LAMPIRAN

Kode Program 4.1 *Splash Screen*

```
using UnityEngine;
using System.Collections;

public class SplashScreenScript : MonoBehaviour {

    public bool Activated;
    public float DelayTime;
    public string NextScene;

    IEnumerator ExecuteSplashScreenScript(){
        if(Activated){
            yield return new WaitForSeconds(DelayTime);
            Application.LoadLevel(NextScene);
        }
    }

    // Use this for initialization
    void Start () {
        StartCoroutine(ExecuteSplashScreenScript());
    }

    // Update is called once per frame
    void Update () {

    }
}

```

Kode Program 4.2 *Main Menu*

```
using UnityEngine;
using System.Collections;

public class MainMenuScript : MonoBehaviour {
    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 500;
        float btnHeight = 100;
        float btnSpace = 1;
        float btnLeft = 980;
        float btnTop = 130;

        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 50),btnWidth,btnHeight),"Penampang Gigi")){
            Application.LoadLevel("Gigi Sub Menu");
        }
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 300),btnWidth,btnHeight),"Struktur Gigi")){
            Application.LoadLevel("Gigi Detail Sub Menu");
        }
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 550),btnWidth,btnHeight),"Gigi Berlubang")){
            Application.LoadLevel("Gigi Berlubang Sub Menu");
        }
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 800),btnWidth,btnHeight),"Quit")){
            Application.Quit();
        }
    }
}
```

Kode Program 4.3 *Button Scan* ke Marker Penampang Gigi

```
using UnityEngine;
using System.Collections;

public class BtnNextGScan : MonoBehaviour {

    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 400;
        float btnHeight = 400;
        float btnSpace = 250;
        float btnLeft = 300;
        float btnTop = -250;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),"Scan Penampang Gigi")){
            Application.LoadLevel("Gigi");
        }
    }
}
```

Kode Program 4.4 *Button* ke Informasi Penampang Gigi

```
using UnityEngine;
using System.Collections;

public class BtnNextGI : MonoBehaviour {

    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 400;
        float btnHeight = 400;
        float btnSpace = 250;
        float btnLeft = 1200;
        float btnTop = -250;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),"Informasi Penampang Gigi")){
            Application.LoadLevel("Gigi Info");
        }
    }
}
```

Kode Program 4.5 *Button* kembali ke *Sub Menu* Penampang Gigi dari Informasi Penampang Gigi

```
using UnityEngine;
using System.Collections;

public class GigiInfoBackObject : MonoBehaviour {

    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 100;
        float btnHeight = 100;
        float btnSpace = 250;
        float btnLeft = 1800;
        float btnTop = -510;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),".")){
            Application.LoadLevel("Gigi Sub Menu");
        }
    }
}
```

Kode Program 4.6 *Button* kembali ke *Main Menu* dari Informasi Penampang Gigi

```
using UnityEngine;
using System.Collections;

public class GigiInfoBackMM : MonoBehaviour {
    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 400;
        float btnHeight = 400;
        float btnSpace = 250;
        float btnLeft = 1500;
        float btnTop = 450;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),"Back to Main Menu")){
            Application.LoadLevel("Gigi Main Menu");
        }
    }
}
```

Kode Program 4.7 *Button* Halaman Informasi Penampang Gigi

```
using UnityEngine;
using System.Collections;

public class BtnNextPrevGI : MonoBehaviour {

    // public Texture2D buttonImage = null;
    public GUISkin MySkin;

    int switchpos = -1;
    bool[] cam = new bool[3];
    Vector3 defualt_pos;

    // Use this for initialization
    public void Start () {
        defualt_pos = transform.position;
    }

    // Update is called once per frame
    public void Update () {
    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    public void OnGUI()
    {
        AutoResize(1920, 1200);
        GUI.skin = MySkin;
        //GUI.backgroundColor = new Color(0, 0, 0, 0);
        cam[0] = (GUI.Button(new Rect(600, 1080, 100, 100), "1"));
        cam[1] = (GUI.Button(new Rect(900, 1080, 100, 100), "2"));
        cam[2] = (GUI.Button(new Rect(1200, 1080, 100, 100), "3"));

        //Switching
        if (cam[0] && switchpos != 0)
        {
            switchpos = 0;
            transform.position = defualt_pos + new Vector3(0, 0, 0);
            //set diaabled button image here
        }
        else if (cam[1] && switchpos != 1)
        {
            switchpos = 1;
            transform.position = defualt_pos + new Vector3(14.5f, 0, 0);
            //set diaabled button image here
        }
        else if (cam[2] && switchpos != 2)
        {
            switchpos = 2;
            transform.position = defualt_pos + new Vector3(29f, 0, 0);
            //set diaabled button image here
        }
    }
}
```

Kode Program 4.8 *Button Scan* ke Marker Struktur Gigi

```
using UnityEngine;
using System.Collections;

public class BtnNextGDScan : MonoBehaviour {

    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 400;
        float btnHeight = 400;
        float btnSpace = 250;
        float btnLeft = 300;
        float btnTop = -250;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),"Scan Struktur Gigi")){
            Application.LoadLevel("Gigi Detail");
        }
    }
}
```

Kode Program 4.9 *Button* ke Informasi Struktur Gigi

```
using UnityEngine;
using System.Collections;

public class BtnNextGDI : MonoBehaviour {

    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 400;
        float btnHeight = 400;
        float btnSpace = 250;
        float btnLeft = 1200;
        float btnTop = -250;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),"Informasi Struktur Gigi")){
            Application.LoadLevel("Gigi Detail Info");
        }
    }
}
```

Kode Program 4.10 *Button* kembali ke ke *Sub Menu* Struktur Gigi dari Informasi Struktur Gigi

```
using UnityEngine;
using System.Collections;

public class GigiDetailInfoBackObject : MonoBehaviour {

    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 100;
        float btnHeight = 100;
        float btnSpace = 250;
        float btnLeft = 1800;
        float btnTop = -510;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),".")){
            Application.LoadLevel("Gigi Detail Sub Menu");
        }
    }
}
```

Kode Program 4.11 *Button* kembali ke *Main Menu* dari Informasi Struktur Gigi

```
using UnityEngine;
using System.Collections;

public class GigiDetailBackMM : MonoBehaviour {
    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 400;
        float btnHeight = 400;
        float btnSpace = 250;
        float btnLeft = 1350;
        float btnTop = 450;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),"Back to Main Menu")){
            Application.LoadLevel("Gigi Main Menu");
        }
    }
}
```

Kode Program 4.12 *Button* Halaman Informasi Struktur Gigi

```
using UnityEngine;
using System.Collections;

public class BtnNextPrevGDI : MonoBehaviour {

    // public Texture2D buttonImage = null;
    public GUISkin MySkin;

    int switchpos = -1;
    bool[] cam = new bool[7];
    Vector3 defualt_pos;

    // Use this for initialization
    public void Start () {
        defualt_pos = transform.position;
    }

    // Update is called once per frame
    public void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    public void OnGUI()
    {
        AutoResize(1920, 1200);
        GUI.skin = MySkin;
        //GUI.backgroundColor = new Color(0, 0, 0, 0);
        cam[0] = (GUI.Button(new Rect(460, 1080, 100, 100), "1"));
        cam[1] = (GUI.Button(new Rect(610, 1080, 100, 100), "2"));
        cam[2] = (GUI.Button(new Rect(760, 1080, 100, 100), "3"));
        cam[3] = (GUI.Button(new Rect(910, 1080, 100, 100), "4"));
        cam[4] = (GUI.Button(new Rect(1060, 1080, 100, 100), "5"));
        cam[5] = (GUI.Button(new Rect(1210, 1080, 100, 100), "6"));
        cam[6] = (GUI.Button(new Rect(1360, 1080, 100, 100), "7"));

        //Switching
        if (cam[0] && switchpos != 0)
        {
            switchpos = 0;
            transform.position = defualt_pos + new Vector3(0, 0, 0);
            //set diaabled button image here
        }
        else if (cam[1] && switchpos != 1)
        {
            switchpos = 1;
            transform.position = defualt_pos + new Vector3(14.5f, 0, 0);
            //set diaabled button image here
        }
         else if (cam[2] && switchpos != 2)
         {
             switchpos = 2;
             transform.position = defualt_pos + new Vector3(29f, 0, 0);
             //set diaabled button image here
         }
         else if (cam[3] && switchpos != 3)
         {
             switchpos = 3;
             transform.position = defualt_pos + new Vector3(0, 0, -8f);
             //set diaabled button image here
         }
         else if (cam[4] && switchpos != 4)
         {
             switchpos = 4;
             transform.position = defualt_pos + new Vector3(14.5f, 0, -8f);
             //set diaabled button image here
         }
         else if (cam[5] && switchpos != 5)
         {
             switchpos = 5;
             transform.position = defualt_pos + new Vector3(29f, 0, -8f);
             //set diaabled button image here
         }
         else if (cam[6] && switchpos != 6)
         {
             switchpos = 6;
             transform.position = defualt_pos + new Vector3(0, 0, -16f);
             //set diaabled button image here
         }
     }
}
```

Kode Program 4.13 *Button Scan* ke Marker Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class BtnNextGBScan : MonoBehaviour {

    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 400;
        float btnHeight = 400;
        float btnSpace = 250;
        float btnLeft = 300;
        float btnTop = -250;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),"Scan Gigi Berlubang")){
            Application.LoadLevel("Gigi Berlubang");
        }
    }
}
```

Kode Program 4.14 *Button* ke Informasi Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class BtnNextGBI : MonoBehaviour {

    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 400;
        float btnHeight = 400;
        float btnSpace = 250;
        float btnLeft = 1200;
        float btnTop = -250;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),"Informasi Gigi Berlubang")){
            Application.LoadLevel("Gigi Berlubang Info");
        }
    }
}
```

Kode Program 4.15 *Button* kembali ke ke *Sub Menu* Gigi Berlubang dari Informasi Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class GigiBerlubangInfoBackObject : MonoBehaviour {

    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 100;
        float btnHeight = 100;
        float btnSpace = 250;
        float btnLeft = 1800;
        float btnTop = -510;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),".")){
            Application.LoadLevel("Gigi Berlubang Sub Menu");
        }
    }
}
```

Kode Program 4.16 *Button* kembali ke *Main Menu* dari Informasi Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class GigiBerlubangInfoBackMM : MonoBehaviour {
    public GUISkin MySkin;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    void OnGUI(){
        AutoResize(1920, 1200);

        float btnWidth = 400;
        float btnHeight = 400;
        float btnSpace = 250;
        float btnLeft = 1500;
        float btnTop = 450;
        GUI.skin = MySkin;
        if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),"Back to Main Menu")){
            Application.LoadLevel("Gigi Main Menu");
        }
    }
}
```

Kode Program 4.17 *Button* kembali ke *Sub Menu* Informasi Gigi Berlubang dari Informasi Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class GigiBerlubangInfoBackMenu : MonoBehaviour {

        public GUISkin MySkin;

        // Use this for initialization
        void Start () {

        }

        // Update is called once per frame
        void Update () {

        }

        public static void AutoResize(int screenWidth, int screenHeight)
        {
            Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
            GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
        }

        void OnGUI(){
            AutoResize(1920, 1200);

        float btnWidth = 100;
        float btnHeight = 100;
        float btnSpace = 250;
        float btnLeft = 1800;
        float btnTop = -510;
            GUI.skin = MySkin;
            if (GUI.Button(new Rect(btnLeft,btnTop + (btnSpace * 2),btnWidth,btnHeight),".")){
                Application.LoadLevel("Gigi Berlubang Info");
            }
        }
    }
```

Kode Program 4.18 *Button* Halaman Informasi Tahapan, Penyebab Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class BtnNextPrevGBI : MonoBehaviour {

    // public Texture2D buttonImage = null;
    public GUISkin MySkin;

    int switchpos = -1;
    bool[] cam = new bool[3];
    Vector3 defualt_pos;

    // Use this for initialization
    public void Start () {
        defualt_pos = transform.position;
    }

    // Update is called once per frame
    public void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    public void OnGUI()
    {
        AutoResize(1920, 1200);
        GUI.skin = MySkin;
        //GUI.backgroundColor = new Color(0, 0, 0, 0);
        cam[0] = (GUI.Button(new Rect(600, 1080, 100, 100), "1"));
        cam[1] = (GUI.Button(new Rect(900, 1080, 100, 100), "2"));
        cam[2] = (GUI.Button(new Rect(1200, 1080, 100, 100), "3"));

        //Switching
        if (cam[0] && switchpos != 0)
        {
            switchpos = 0;
            transform.position = defualt_pos + new Vector3(0, 0, 0);
            //set diaabled button image here
        }
        else if (cam[1] && switchpos != 1)
        {
            switchpos = 1;
            transform.position = defualt_pos + new Vector3(14.5f, 0, 0);
            //set diaabled button image here
        }
        else if (cam[2] && switchpos != 2)
        {
            switchpos = 2;
            transform.position = defualt_pos + new Vector3(29f, 0, 0);
            //set diaabled button image here
        }
    }
}
```

Kode Program 4.19 *Button* Halaman Informasi Mencegah Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class BtnNextPrevGBIsatu : MonoBehaviour {

    // public Texture2D buttonImage = null;
    public GUISkin MySkin;

    int switchpos = -1;
    bool[] cam = new bool[6];
    Vector3 defualt_pos;

    // Use this for initialization
    public void Start () {
        defualt_pos = transform.position;
    }

    // Update is called once per frame
    public void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    public void OnGUI()
    {
        AutoResize(1920, 1200);
        GUI.skin = MySkin;
        //GUI.backgroundColor = new Color(0, 0, 0, 0);
        cam[0] = (GUI.Button(new Rect(500, 1080, 100, 100), "1"));
        cam[1] = (GUI.Button(new Rect(650, 1080, 100, 100), "2"));
        cam[2] = (GUI.Button(new Rect(800, 1080, 100, 100), "3"));
        cam[3] = (GUI.Button(new Rect(950, 1080, 100, 100), "4"));
        cam[4] = (GUI.Button(new Rect(1100, 1080, 100, 100), "5"));
        cam[5] = (GUI.Button(new Rect(1250, 1080, 100, 100), "6"));

        //Switching
        if (cam[0] && switchpos != 0)
        {
            switchpos = 0;
            transform.position = defualt_pos + new Vector3(0, 0, 0);
            //set diaabled button image here
        }
        else if (cam[1] && switchpos != 1)
        {
            switchpos = 1;
            transform.position = defualt_pos + new Vector3(14.5f, 0, 0);
            //set diaabled button image here
        }
        else if (cam[2] && switchpos != 2)
        {
           switchpos = 2;
           transform.position = defualt_pos + new Vector3(29f, 0, 0);
           //set diaabled button image here
        }
        else if (cam[3] && switchpos != 3)
        {
            switchpos = 3;
            transform.position = defualt_pos + new Vector3(0, 0, -8f);
            //set diaabled button image here
        }
        else if (cam[4] && switchpos != 4)
        {
            switchpos = 4;
            transform.position = defualt_pos + new Vector3(14.5f, 0, -8f);
            //set diaabled button image here
        }
        else if (cam[5] && switchpos != 5)
        {
            switchpos = 5;
            transform.position = defualt_pos + new Vector3(29f, 0, -8f);
            //set diaabled button image here
        }
    }
}
```

Kode Program 4.20 *Button* Halaman Informasi Pengertian, Merawat Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class BtnNextPrevGBIdua : MonoBehaviour {

    // public Texture2D buttonImage = null;
    public GUISkin MySkin;

    int switchpos = -1;
    bool[] cam = new bool[2];
    Vector3 defualt_pos;

    // Use this for initialization
    public void Start () {
        defualt_pos = transform.position;
    }

    // Update is called once per frame
    public void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    public void OnGUI()
    {
        AutoResize(1920, 1200);
        GUI.skin = MySkin;
        //GUI.backgroundColor = new Color(0, 0, 0, 0);
        cam[0] = (GUI.Button(new Rect(750, 1080, 100, 100), "1"));
        cam[1] = (GUI.Button(new Rect(1050, 1080, 100, 100), "2"));

        //Switching
        if (cam[0] && switchpos != 0)
        {
            switchpos = 0;
            transform.position = defualt_pos + new Vector3(0, 0, 0);
            //set diabled button image here
        }
        else if (cam[1] && switchpos != 1)
        {
            switchpos = 1;
            transform.position = defualt_pos + new Vector3(14.5f, 0, 0);
            //set diabled button image here
        }
    }
}
```

Kode Program 4.21 *Button* Halaman Informasi Akibat Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class BtnNextPrevGBItiga : MonoBehaviour {

    // public Texture2D buttonImage = null;
    public GUISkin MySkin;

    int switchpos = -1;
    bool[] cam = new bool[4];
    Vector3 defualt_pos;

    // Use this for initialization
    public void Start () {
        defualt_pos = transform.position;
    }

    // Update is called once per frame
    public void Update () {

    }

    public static void AutoResize(int screenWidth, int screenHeight)
    {
        Vector2 resizeRatio = new Vector2((float)Screen.width / screenWidth, (float)Screen.height / screenHeight);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new Vector3(resizeRatio.x, resizeRatio.y, 1.0f));
    }

    public void OnGUI()
    {
        AutoResize(1920, 1200);
        GUI.skin = MySkin;
        //GUI.backgroundColor = new Color(0, 0, 0, 0);
        cam[0] = (GUI.Button(new Rect(600, 1080, 100, 100), "1"));
        cam[1] = (GUI.Button(new Rect(800, 1080, 100, 100), "2"));
        cam[2] = (GUI.Button(new Rect(1000, 1080, 100, 100), "3"));
        cam[3] = (GUI.Button(new Rect(1200, 1080, 100, 100), "4"));

        //Switching
        if (cam[0] && switchpos != 0)
        {
            switchpos = 0;
            transform.position = defualt_pos + new Vector3(0, 0, 0);
            //set diaabled button image here
        }
        else if (cam[1] && switchpos != 1)
        {
            switchpos = 1;
            transform.position = defualt_pos + new Vector3(14.5f, 0, 0);
            //set diaabled button image here
        }
        else if (cam[2] && switchpos != 2)
        {
            switchpos = 2;
            transform.position = defualt_pos + new Vector3(29f, 0, 0);
            //set diaabled button image here
        }
        else if (cam[3] && switchpos != 3)
        {
            switchpos = 3;
            transform.position = defualt_pos + new Vector3(43.5f, 0, 0);
            //set diaabled button image here
        }
    }
}
```

Kode Program 4.22 *Virtual Buttons* pada Marker Gigi Berlubang

```
using UnityEngine;
using Vuforia;
using System.Collections;

public class VBGigiBerlubang : MonoBehaviour, IVirtualButtonEventHandler {
    private GameObject gigi_plak;
    private GameObject gigi_bolong1;
    private GameObject gigi_bolong2;
    private GameObject gigi_rusak;

    // Use this for initialization
    void Start () {
        VirtualButtonBehaviour[] vbs = transform.GetComponentsInChildren<VirtualButtonBehaviour> ();

        for (int i=0; i < vbs.Length; ++i) {
            vbs[i].RegisterEventHandler(this);
    }
        // Find the models based on the names in the Hierarchy
        gigi_plak = transform.FindChild("gigi_plak").gameObject;
        gigi_bolong1 = transform.FindChild("gigi_bolong1").gameObject;
        gigi_bolong2 = transform.FindChild("gigi_bolong2").gameObject;
        gigi_rusak = transform.FindChild("gigi_rusak").gameObject;

        // We don't want to show some during the startup
        gigi_bolong1.SetActive(false);
        gigi_bolong2.SetActive(false);
        gigi_rusak.SetActive(false);

    }

    public void OnButtonPressed(VirtualButtonAbstractBehaviour vb){
        //specify which button you want to function by using the if statement
        switch(vb.VirtualButtonName) {
        case "btnplakgigi":
            gigi_plak.SetActive(false);
            gigi_bolong1.SetActive(true);
            gigi_bolong2.SetActive(false);
            gigi_rusak.SetActive(false);
            break;
        case "btnkaranggigi":
            gigi_plak.SetActive(false);
            gigi_bolong1.SetActive(false);
            gigi_bolong2.SetActive(true);
            gigi_rusak.SetActive(false);
            break;
        case "btngigiberlubang":
            gigi_plak.SetActive(false);
            gigi_bolong1.SetActive(false);
            gigi_bolong2.SetActive(false);
            gigi_rusak.SetActive(true);
            break;
        default:
            throw new UnityException("Button not supported: " + vb.VirtualButtonName);
            break;
        }
    }

    public void OnButtonReleased(VirtualButtonAbstractBehaviour vb){
        //specify which button you want to function by using the if statement
        switch(vb.VirtualButtonName) {
        case "btnplakgigi":
            gigi_plak.SetActive(false);
            gigi_bolong1.SetActive(true);
            gigi_bolong2.SetActive(false);
            gigi_rusak.SetActive(false);
            break;
        case "btnkaranggigi":
            gigi_plak.SetActive(false);
            gigi_bolong1.SetActive(false);
            gigi_bolong2.SetActive(true);
            gigi_rusak.SetActive(false);
            break;
        case "btngigiberlubang":
            gigi_plak.SetActive(true);
            gigi_bolong1.SetActive(false);
            gigi_bolong2.SetActive(false);
            gigi_rusak.SetActive(false);
            break;
        default:
            throw new UnityException("Button not supported: " + vb.VirtualButtonName);
            break;
        }
    }

    // Update is called once per frame
    void Update () {

    }
}
```

Kode Program 4.23 *Register Virtual Buttons* dan Marker (*Image Target*) pada xml file di AR_Gigi *package*

```
<?xml version="1.0" encoding="UTF-8"?>
<QCARConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="qcar_config.xsd">
  <Tracking>
    <ImageTarget name="GigiBerlubang" size="50.000000 50.000000" >
        <VirtualButton name="btnplakgigi" rectangle="-0.4379999 0.001000001 -0.236 -0.4119999" enabled="true"/>
        <VirtualButton name="btnkaranggigi" rectangle="-0.09799998 0.001000001 0.09599997 -0.4119999" enabled="true"/>
        <VirtualButton name="btngigiberlubang" rectangle="0.24 0.001000001 0.4249999 -0.4119999" enabled="true"/>
    </ImageTarget>
    <ImageTarget name="StrukturGigi" size="500.000000 500.000000" />
    <ImageTarget name="Gigi" size="50.000000 50.000000" />
  </Tracking>
</QCARConfig>
```

Kode Program 4.24 *Touch Logic* ke Informasi Penampang Gigi untuk objek Penampang Gigi

```
using UnityEngine;
using System.Collections;

public class TouchLogic : MonoBehaviour
{
    public static int currTouch = 0;//so other scripts can know what touch is currently on screen
    private Ray ray;//this will be the ray that we cast from our touch into the scene
    private RaycastHit rayHitInfo = new RaycastHit();//return the info of the object that was hit by the ray
    [HideInInspector]
    public int touch2Watch = 64;
    void Update ()
    {
        //is there a touch on screen?
        if(Input.touches.Length <= 0)
        {
            //if no touches then execute this code
        }
        else //if there is a touch
        {
            //loop through all the the touches on screen
            for(int i = 0; i < Input.touchCount; i++)
            {
                currTouch = i;
                Debug.Log(currTouch);
                //executes this code for current touch (i) on screen
                if(this.guiTexture != null && (this.guiTexture.HitTest(Input.GetTouch(i).position)))
                {
                    //if current touch hits our guitexture, run this code
                    if(Input.GetTouch(i).phase == TouchPhase.Began)
                    {
                        //need to send message b/c function is not present in this script
                        //OnTouchBegan();
                        this.SendMessage("OnTouchBegan");
                    }
                    if(Input.GetTouch(i).phase == TouchPhase.Ended)
                    {
                        //OnTouchEnded();
                        this.SendMessage("OnTouchEnded");
                    }
                    if(Input.GetTouch(i).phase == TouchPhase.Moved)
                    {
                        //OnTouchMoved();
                        this.SendMessage("OnTouchMoved");
                    }
                    //if(Input.GetTouch(i).phase == TouchPhase.Stationary)
                    //{
                        //OnTouchStayed();
                    //  this.SendMessage("OnTouchStayed");
                    //}
                }
                //outside so it doesn't require the touch to be over the guitexture
                if(Input.GetTouch(i).phase == TouchPhase.Began)
                {
                    //need to send message b/c function is not present in this script
                    //OnTouchBegan();
                    this.SendMessage("OnTouchBeganAnywhere");
                }
                if(Input.GetTouch(i).phase == TouchPhase.Ended)
                {
                    //OnTouchEnded();
                    this.SendMessage("OnTouchEndedAnywhere");
                }
                if(Input.GetTouch(i).phase == TouchPhase.Moved)
                {
                    //OnTouchMoved();
                    this.SendMessage("OnTouchMovedAnywhere");
                }
                if(Input.GetTouch(i).phase == TouchPhase.Stationary)
                {
                    //OnTouchStayed();
                    this.SendMessage("OnTouchStayedAnywhere");
                }

                //this is for 3d object with colliders
                if(Input.GetTouch(i).phase == TouchPhase.Began)
                {
                    ray = Camera.mainCamera.ScreenPointToRay(Input.GetTouch(i).position);//creates ray from screen point position
                    if(Physics.Raycast(ray, out rayHitInfo))
                    {
                        Application.LoadLevel("Gigi Info");
                        rayHitInfo.transform.gameObject.SendMessage("OnTouchBegan3D");
                    }
                    // ray = Camera.mainCamera.ScreenPointToRay(Input.GetTouch(i).position);//creates ray from screen point position
                }
            }
        }
    }
}
```

Kode Program 4.25 *Touch Logic* ke Informasi Struktur Gigi untuk objek Struktur Gigi

```
using UnityEngine;
using System.Collections;

public class TouchLogicGD : MonoBehaviour {

    public static int currTouch = 0;//so other scripts can know what touch is currently on screen
    private Ray ray;//this will be the ray that we cast from our touch into the scene
    private RaycastHit rayHitInfo = new RaycastHit();//return the info of the object that was hit by the ray
    [HideInInspector]
    public int touch2Watch = 64;
    void Update ()
    {
        //is there a touch on screen?
        if(Input.touches.Length <= 0)
        {
            //if no touches then execute this code
        }
        else //if there is a touch
        {
            //loop through all the the touches on screen
            for(int i = 0; i < Input.touchCount; i++)
            {
                currTouch = i;
                Debug.Log(currTouch);
                //executes this code for current touch (i) on screen
                if(this.guiTexture != null && (this.guiTexture.HitTest(Input.GetTouch(i).position)))
                {
                    //if current touch hits our guitexture, run this code
                    if(Input.GetTouch(i).phase == TouchPhase.Began)
                    {
                        //need to send message b/c function is not present in this script
                        //OnTouchBegan();
                        this.SendMessage("OnTouchBegan");
                    }
                    if(Input.GetTouch(i).phase == TouchPhase.Ended)
                    {
                        //OnTouchEnded();
                        this.SendMessage("OnTouchEnded");
                    }
                    if(Input.GetTouch(i).phase == TouchPhase.Moved)
                    {
                        //OnTouchMoved();
                        this.SendMessage("OnTouchMoved");
                    }
                    //if(Input.GetTouch(i).phase == TouchPhase.Stationary)
                    //{
                    //OnTouchStayed();
                    //  this.SendMessage("OnTouchStayed");
                    //}
                }
                //outside so it doesn't require the touch to be over the guitexture
                if(Input.GetTouch(i).phase == TouchPhase.Began)
                {
                    //need to send message b/c function is not present in this script
                    //OnTouchBegan();
                    this.SendMessage("OnTouchBeganAnywhere");
                }
                if(Input.GetTouch(i).phase == TouchPhase.Ended)
                {
                    //OnTouchEnded();
                    this.SendMessage("OnTouchEndedAnywhere");
                }
                if(Input.GetTouch(i).phase == TouchPhase.Moved)
                {
                    //OnTouchMoved();
                    this.SendMessage("OnTouchMovedAnywhere");
                }
                if(Input.GetTouch(i).phase == TouchPhase.Stationary)
                {
                    //OnTouchStayed();
                    this.SendMessage("OnTouchStayedAnywhere");
                }

                //this is for 3d object with colliders
                if(Input.GetTouch(i).phase == TouchPhase.Began)
                {
                    ray = Camera.mainCamera.ScreenPointToRay(Input.GetTouch(i).position);//creates ray from screen point position
                    if(Physics.Raycast(ray, out rayHitInfo))
                    {
                        Application.LoadLevel("Gigi Detail Info");
                        rayHitInfo.transform.gameObject.SendMessage("OnTouchBegan3D");
                    }
                    // ray = Camera.mainCamera.ScreenPointToRay(Input.GetTouch(i).position);//creates ray from screen point position
                }
            }
        }
    }
}
```

Kode Program 4.26 *Touch Logic* ke Informasi Gigi Berlubang untuk objek Gigi Berlubang

```
using UnityEngine;
using System.Collections;

public class TouchLogicGB : MonoBehaviour {
    public static int currTouch = 0;//so other scripts can know what touch is currently on screen
    private Ray ray;//this will be the ray that we cast from our touch into the scene
    private RaycastHit rayHitInfo = new RaycastHit();//return the info of the object that was hit by the ray
    [HideInInspector]
    public int touch2Watch = 64;
    void Update ()
    {
        //is there a touch on screen?
        if(Input.touches.Length <= 0)
        {
            //if no touches then execute this code
        }
        else //if there is a touch
        {
            //loop through all the the touches on screen
            for(int i = 0; i < Input.touchCount; i++)
            {
                currTouch = i;
                Debug.Log(currTouch);
                //executes this code for current touch (i) on screen
                if(this.guiTexture != null && (this.guiTexture.HitTest(Input.GetTouch(i).position)))
                {
                    //if current touch hits our guitexture, run this code
                    if(Input.GetTouch(i).phase == TouchPhase.Began)
                    {
                        //need to send message b/c function is not present in this script
                        //OnTouchBegan();
                        this.SendMessage("OnTouchBegan");
                    }
                    if(Input.GetTouch(i).phase == TouchPhase.Ended)
                    {
                        //OnTouchEnded();
                        this.SendMessage("OnTouchEnded");
                    }
                    if(Input.GetTouch(i).phase == TouchPhase.Moved)
                    {
                        //OnTouchMoved();
                        this.SendMessage("OnTouchMoved");
                    }
                    //if(Input.GetTouch(i).phase == TouchPhase.Stationary)
                    //{
                    //OnTouchStayed();
                    //  this.SendMessage("OnTouchStayed");
                    //}
                }
                //outside so it doesn't require the touch to be over the guitexture
                if(Input.GetTouch(i).phase == TouchPhase.Began)
                {
                    //need to send message b/c function is not present in this script
                    //OnTouchBegan();
                    this.SendMessage("OnTouchBeganAnywhere");
                }
                if(Input.GetTouch(i).phase == TouchPhase.Ended)
                {
                    //OnTouchEnded();
                    this.SendMessage("OnTouchEndedAnywhere");
                }
                if(Input.GetTouch(i).phase == TouchPhase.Moved)
                {
                    //OnTouchMoved();
                    this.SendMessage("OnTouchMovedAnywhere");
                }
                if(Input.GetTouch(i).phase == TouchPhase.Stationary)
                {
                    //OnTouchStayed();
                    this.SendMessage("OnTouchStayedAnywhere");
                }

                //this is for 3d object with colliders
                if(Input.GetTouch(i).phase == TouchPhase.Began)
                {
                    ray = Camera.mainCamera.ScreenPointToRay(Input.GetTouch(i).position);//creates ray from screen point position
                    if(Physics.Raycast(ray, out rayHitInfo))
                    {
                        Application.LoadLevel("Gigi Berlubang Info");
                        rayHitInfo.transform.gameObject.SendMessage("OnTouchBegan3D");
                    }
                    // ray = Camera.mainCamera.ScreenPointToRay(Input.GetTouch(i).position);//creates ray from screen point position
                }
            }
        }
    }
}
```