



Analisis Kualitas *Learning Management System* Universitas Negeri Makassar Menggunakan Metode Metriks Kualitas Perangkat Lunak

Hasrul Bakri¹, Firdaus², Dyah Vitalocca³

Universitas Negeri Makassar

Email: hasrulbakri@unm.ac.id

Abstrak. Penelitian ini bertujuan untuk mengetahui kualitas *Learning Managements System* (LMS) Universitas Negeri Makassar menggunakan metode metriks kualitas perangkat lunak. Responden pada penelitian ini adalah mahasiswa dan dosen Universitas Negeri Makassar yang berjumlah 250 orang. Data dikumpulkan menggunakan kuisisioner. Data dianalisis menggunakan metode pembobotan berdasarkan metode metriks kualitas perangkat lunak ISO/IEC 25010. Nilai kriteria per faktora dan nilai akhir menggunakan *weight summation*. Nilai *w* adalah nilai metriks yang diperoleh dari hasil pengolahan kuisisioner dan *m* adalah bobot dari sub faktor atau faktor yang sudah diperoleh dari hasil perhitungan dengan menggunakan metode *Rank Order Centroid* (ROC). Berdasarkan hasil penelitian disimpulkan: 1) faktor *functionality* (fungsionalitas) dengan nilai 0,48 dalam kategori *medium* atau cukup, 2) faktor *Usability* dengan nilai 0,54 dalam kategori *medium* atau cukup, 3) faktor *efficiency* dengan nilai 0,53 atau masuk ke dalam kategori *medium* yang berarti perangkat lunak ini cukup efisien, 4) faktor *maintability* dengan nilai 0,21 yang berarti perangkat lunak ini termasuk dalam kategori *low*, 5) faktor *Portability* dengan nilai 0,55 yang berarti perangkat lunak ini termasuk dalam kategori *medium* atau cukup, 6) nilai akhir kualitas eksternal yaitu 0,45 termasuk ke dalam kategori *medium* atau cukup.

Kata Kunci: LMS, Metriks kualitas perangkat lunak, ISO/IEC25010

PENDAHULUAN

McConnell's *Code Complete* (2004) membagi perangkat lunak ke dalam dua hal yaitu *internal* dan *external quality characteristics*. Karakteristik kualitas eksternal merupakan bagian-bagian dari suatu produk yang berhubungan dengan para pemakainya, sedangkan karakteristik kualitas internal tidak secara langsung berhubungan dengan pemakai. *Software quality* didefinisikan sebagai kesesuaian yang diharapkan pada semua *software* yang dibangun dalam hal fungsi *software* yang diutamakan dan unjuk kerja *software*, standar pembangunan *software* yang terdokumentasi dan karakteristik yang ditunjukkan oleh *software* (ISO/IEC 25010, 2011). Definisi ini menekankan pada tiga hal yaitu (Laporte, & April 2018):

1. Kebutuhan *software* adalah fondasi ukuran kualitas *software*, jika *software* Tidak sesuai dengan kebutuhan yang ditentukan maka kualitas pun kurang.
2. Jika menggunakan suatu standar untuk pembangunan *software* maka jika *software* tidak memenuhi standar tersebut maka dianggap kurang berkualitas.
3. Seringkali ada kualitas yang secara langsung diutarakan (tersirat) seperti kemudahan penggunaan dan pemeliharaan yang baik. Kualitas *software* dipertanyakan jika tidak memenuhi kebutuhan ini.

Menurut ISO/IEC 25010 (2011), model metrik kualitas perangkat lunak dalam keilmuan rekayasa perangkat lunak terdapat beberapa model kualitas perangkat lunak. Pada setiap model ini terdapat beberapa faktor yang menjadi poin-poin utama dalam penilaian kualitas sebuah perangkat lunak. Berikut adalah penjelasan tentang model kualitas perangkat lunak yang akan diterapkan. Model ISO/IEC 25010 dikenalkan pertama kali pada tahun 2011 sebagai standarisasi kualitas produk perangkat lunak. Standarisasi ini dibuat karena banyaknya model kualitas yang ditawarkan sebagai faktor kualitas perangkat lunak. Dalam dokumen pertama model ISO/IEC 25010 terdiri dari beberapa bagian model kualitas untuk kualitas sebuah produk perangkat lunak, di antaranya: *internal quality*, *external quality model* dan *quality in use model*

Model kualitas eksternal bagian pertama dari kualitas model tersebut menentukan delapan karakteristik yaitu *portability*, *performance efficiency*, *reliability*, *security*, *usability*, *maintainability*, *compatibility*, dan *functional Suitability*. Berikut adalah karakteristik dan sub karakteristik dari *product quality* (www.iso25000.com):

1. *Functional Suitability* merupakan kemampuan suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan bila digunakan dalam kondisi tertentu. Terdapat tiga sub subkarakteristik, yaitu: *functional completeness*, *functional correctness* dan *functional appropriateness*.
2. *Performance efficiency* merupakan kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi lain. Terdapat tiga subkarakteristik, yaitu: *time Behaviour*, *resource utilization* dan *capacity*.
3. *Compatibility* berkaitan dengan kemampuan produk dapat bertukar informasi dengan produk lain. Terdapat sub-karakteristik, yaitu *co-existence* dan *interoperability*.
4. *Usability* adalah kemampuan produk dapat digunakan oleh pengguna untuk mencapai tujuan tertentu dengan efektivitas, efisiensi dan kepuasan dalam konteks tertentu. Terdapat enam subkarakteristik, yaitu: *appropriateness*, *recognizability*, *learnability*, *operability*, *user error protection*, *user interface aesthetics* dan *accessibility*.
5. *Reliability* merupakan kemampuan sistem menjalankan fungsionalitas pada beberapa kondisi tertentu. Terdapat empat sub-karakteristik, yaitu: *maturity*, *availability*, *fault tolerance*, dan *discoverability*.

6. *Security* memampukan produk melindungi informasi dan data. Terdapat empat subkarakteristik, yaitu: *confidentiality*, *integrity*, *non-repudiation*, *ccountability*, dan *authenticity*.
7. *Maintainability* merupakan tingkat efektivitas dan efisiensi suatu produk dapat dimodifikasi untuk diperbaiki, diperbaiki atau disesuaikan dengan perubahan lingkungan dan kebutuhan. Terdapat lima subkarakteristik, yaitu: *modularity*, *reusability*, *analysability*, *modifiability*, dan *testability*
8. *Portability* berkaitan dengan kemampuan produk dapat dikirim ke lingkungan yang berbeda. Terdapat tiga subkarakteristik, yaitu: *adaptability*, *installability* dan *replaceability*.

Metode Pengukuran Kualitas Perangkat Lunak

Pengukuran kualitas perangkat lunak merupakan hal yang terpenting dari praktek rekayasa perangkat lunak yang baik serta pengukuran aktivitas manusia menambah nilai dan terlibat aktif dalam proses informasi. Pengukuran ini membantu untuk membuat karakteristik khusus dari proses dan produk informasi. Pengukuran meliputi evaluasi kuantitatif yang biasanya menggunakan metrik serta ukuran dapat menggunakan secara langsung menentukan pencapaian tujuan kualitas numerik.

Pengukuran selalu menjadi fundamental bagi kemajuan untuk setiap disiplin rekayasa dan pengujian perangkat lunak. pengukuran perangkat lunak dan metrik menjadi peran mendasar dalam siklus hidup pengembangan perangkat lunak, dan metrik perangkat lunak telah digunakan dalam membuat keputusan kuantitatif/kualitatif maupun dalam penilaian risiko dan pengurangan dalam proyek perangkat lunak. Proses pengukuran merupakan cara untuk melakukan pengukuran faktor kualitas dari sebuah perangkat lunak. Pengukuran kualitas perangkat lunak akan menggunakan metrik ISO/IEC 25010. Metrik 25010 ini melakukan perhitungan internal dan external berdasarkan karakteristik dari delapan faktor yang ada pada model ISO/IEC 25010.

Metrik internal dapat digunakan untuk membantu memprediksi apakah perangkat lunak akan memenuhi kebutuhan keandalan organisasi selama pengembangan sistem perangkat lunak. *Maturity metrik* menunjukkan satu set atribut untuk menilai kematangan suatu perangkat lunak. Deteksi kesalahan adalah sebuah ukuran seberapa banyak *defects* terdeteksi dalam produk perangkat lunak. Metrik ini dikumpulkan dengan menghitung jumlah terdeteksi *bug* yang ditemukan dalam *review* dan membandingkan angka itu dengan jumlah yang diperkirakan dapat ditemukan dalam tahap pengujian statis. Metrik yang dihitung oleh rumus $X = A / B$, dimana A adalah jumlah absolut *bug* terdeteksi (dari *review* laporan) dan B adalah perkiraan jumlah *expected*. Nilai tinggi untuk metrik ini menyiratkan seberapa baik tingkat kualitas produk.

Penghapusan kesalahan adalah sebuah ukuran seberapa banyak cacat yang ditemukan selama *review* dihapus (dikoreksi) selama desain dan implementasi tahap.

Metrik dapat dihitung dengan rumus $X = A / B$. Dimana A adalah jumlah bug tetap selama desain dan coding dan B adalah Jumlah yang ditemukan selama *review*. Semakin dekat nilai adalah 1, yang lebih baik. A Nilai penghapusan kesalahan orang akan berarti bahwa setiap cacat terdeteksi telah dihapus. Metriks eksternal ISO/IEC 25010 mempunyai beberapa metrik eksternal. Metriks eksternal didefinisikan dalam ISO/IEC 25010 dan diukur selama pengujian dinamis. Kategori metrik ini sesuai dengan yang internal yaitu *maturity, fault tolerance, recoverability, and compliance*. Metriks kematangan mengukur kebebasan sistem perangkat lunak dari kegagalan disebabkan oleh *bug* di sistem itu sendiri. Faktor eksternal ini lebih menilai dari segi program kepada pengguna sedangkan internal lebih menilai dari segi program tersebut.

Perhitungan Metriks pada metriks ISO/IEC 25010 terdapat 8 faktor dan beberapa subfaktor yang akan digunakan sebagai perhitungan yang tentunya akan dilihat dari karakteristik perangkat lunak tersebut. Setiap faktor karakteristik mempunyai hitungan tersendiri yang sudah ditentukan oleh metriks ISO/IEC 25010.

Tabel 1. Subfaktor Metriks

<i>Functionality Metrics</i>	<i>Suitability metrics</i> <i>Accuracy metrics</i> <i>InterOperability metrics</i> <i>Security metrics</i> <i>Functionality compliance metrics</i>
<i>Reliability metrics</i>	<i>Maturity metrics</i> <i>Fault tolerance metrics</i> <i>Recoverability metrics</i> <i>Reliability compliance metrics</i>
<i>Usability Metrics</i>	<i>Understandability metrics</i> <i>Learnability metrics</i> <i>Operability metrics</i> <i>Attractiveness metrics</i> <i>Usability compliance metrics</i>
<i>Efficiency metrics</i>	<i>Time Behaviour metrics</i> <i>Resource utilisation metrics</i> <i>Efficiency compliance metrics</i>
<i>Maintainability metrics</i>	<i>Analysability metrics</i> <i>Changeability metrics</i> <i>Stability metrics</i> <i>Testability metrics</i> <i>Maintainability compliance metrics</i>
<i>Portability metrics</i>	<i>Adaptability metrics</i> <i>Installability metrics</i>

	<i>Co-existence metrics</i>
	<i>Replaceability metrics</i>
	<i>Portability compliance metrics</i>
<i>Security</i>	<i>Confidentiality metrics</i>
	<i>Integrity metrics</i>
	<i>Non-repudiation metrics</i>
	<i>Accountability metrics</i>
	<i>Authenticity metrics</i>
<i>Compatibility</i>	<i>Co-existence metrics</i>
	<i>InterOperability metrics</i>

Pengukuran perangkat lunak adalah faktor utama untuk mengetahui seberapa baik perangkat lunak yang akan dinilai berdasarkan metode yang digunakan. Pengukuran perangkat lunak dilakukan dengan cara mencari nilai kriteria perangkat lunak tersebut dan juga mencari bobot kepentingan berdasarkan metode yang akan digunakan. Penilaian keseluruhan dapat diperoleh dari hasil nilai kriteria dan juga nilai bobot. Perhitungan bobot dan nilai kriteria bisa bervariasi tergantung dengan metode apa yang digunakan.

Metode Pembobotan

Rank Order Centroid (ROC) didasarkan pada tingkat kepentingan atau prioritas dari kriteria. Menurut Jeffreys dan Cockfield (2008) dalam Rahmah (2013), teknik ROC memberikan bobot pada setiap kriteria sesuai dengan ranking yang dinilai berdasarkan tingkat prioritas. Biasanya dibentuk dengan pernyataan “Kriteria 1 lebih penting dari kriteria 2, yang lebih penting dari kriteria 3” dan seterusnya hingga kriteria ke n, ditulis . Untuk menentukan bobotnya, diberikan aturan yang sama yaitu di mana merupakan bobot untuk kriteria. Atau dapat dijelaskan sebagai berikut:

Jika $Cr_1 \geq Cr_2 \geq Cr_3 \geq \dots \geq Cr_n$

Maka $W_1 \geq W_2 \geq W_3 \geq \dots \geq W_n$

Selanjutnya, jika k merupakan banyaknya kriteria, maka

$$W_1 = \frac{1 + \frac{1}{2} + \frac{1}{3} + \dots + 1/k}{k}$$

$$W_2 = \frac{0 + \frac{1}{2} + \frac{1}{3} + \dots + 1/k}{k}$$

$$W_3 = \frac{0 + 0 + \frac{1}{3} + \dots + 1/k}{k}$$

$$W_k = \frac{0 + 0 + 0 + \dots + 1/k}{k}$$

Secara umum pembobotan ROC dapat dirumuskan sebagai berikut :

$$W_k = 1/k \sum_{i=1}^k (1/i)$$

Dimana :

- k = Banyaknya komponen
- i = Bilangan bulat 0 ke $-k$
- W_k = Bobot untuk komponen ke- i

Metode Pengolahan Kuesioner

Kuesioner merupakan sekumpulan pertanyaan yang diajukan kepada responden dengan tujuan dan maksud tertentu untuk memperoleh data, sehingga data tersebut dapat digunakan atau diolah kembali. Untuk mengolah data dari kuesioner tersebut dapat menggunakan berbagai macam metode pengolahan data sesuai dengan rumus metrik yang digunakan pada ISO/IEC 52010. Kuisisioner ini diajukan untuk memenuhi nilai A, B maupun T yang dibutuhkan untuk menghitung nilai metrik dari setiap kriteria yang sudah ada. Dari hasil jawaban yang diberikan oleh responden akan dicari nilai rata-rata, hasil jawaban responden akan dijumlahkan dan totalnya akan dibagi dengan jumlah responden. Maka akan didapatkan nilai A, B ataupun T.

METODE PENELITIAN

Data dalam penelitian ini dikumpulkan dengan menggunakan kuisisioner. Responden pada penelitian ini terdiri atas dosen dan mahasiswa yang berjumlah 250 orang. Perhitungan nilai sub faktor, faktor maupun nilai keseluruhan menjadi faktor penting untuk mengetahui hasil akhir sebuah nilai. Menentukan nilai kriteria per faktornya dan nilai akhir dengan rumus perhitungan *Weight Summation*. Nilai w adalah nilai metrik yang diperoleh dari hasil pengolahan kuisisioner dan m adalah bobot dari sub faktor atau faktor yang sudah diperoleh dari hasil perhitungan dengan menggunakan metode *Rank Order Centroid* (ROC).

HASIL DAN PEMBAHASAN

Pengujian dilakukan menggunakan standard kualitas perangkat lunak ISO/IEC 25010. Dengan kriteria metrik yang terdapat pada model ISO/IEC 25010:

1. *Functionality* (fungsionalitas), perangkat lunak mampu menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu. *Functionality* terdiri atas tiga sub karakteristik, yaitu:
 - a. *Accuracy*, perangkat lunak ini mampu memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.
 - b. *Suitability*, perangkat lunak ini mampu menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna.
 - c. *Compliance*, kemampuan perangkat lunak dalam memenuhi standard an kebutuhan sesuai peraturan yang berlaku
 - d. *InterOperability*, perangkat lunak ini mampu untuk berinteraksi dengan satu atau lebih sistem tertentu

Tabel 2. Pembobotan *Functionality*

Sub Faktor	Bobot
<i>Accuracy</i>	$W1 = \frac{1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}}{4} = 0.52$
<i>Suitability</i>	$W2 = \frac{0+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}}{4} = 0.27$
<i>Compliance</i>	$W3 = \frac{0+0+\frac{1}{3}+\frac{1}{4}}{4} = 0.15$
<i>InterOperability</i>	$W4 = \frac{0+0+0+\frac{1}{4}}{4} = 0.06$

Nilai *Functionality* = $(w_{accuracy} * m_{accuracy}) + (w_{Suitability} * m_{Suitability}) + (w_{compliance} * m_{compliance}) + (w_{interOperability} * m_{interOperability})$

Nilai *Functionality* = $(0,35 * 0,52) + (0,7 * 0,27) + (0,50 * 0,15) + (0,56 * 0,06) = 0.18 + 0.19 + 0.08 + 0.03 = 0.48$

Dari segi *Functionality* (fungsionalitas) dengan hasil 0.48 Perangkat lunak ini termasuk ke dalam kategori *medium*.

1. *Usability* (kebergunaan), perangkat lunak ini dapat untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu. *Usability* terdiri atas dua sub karakteristik, yaitu:
 - a. *Understandbility*, perangkat lunak mudah dipahami.
 - b. *Attractive*, kemampuan perangkat lunak dalam menarik pengguna.
 - c. *Operability*, perangkat lunak ini mudah untuk dioperasikan.

Tabel 3. Pembobotan Usability

Sub Faktor	Bobot
<i>Understandbility</i>	$W1 = \frac{1+\frac{1}{2}+\frac{1}{3}}{3} = 0.61$
<i>Attractive</i>	$W2 = \frac{1+\frac{1}{2}+\frac{1}{3}}{3} = 0.28$
<i>Operability</i>	$W3 = \frac{0+0+\frac{1}{3}}{3} = 0.11$

Nilai *Suitability* $(w \text{ understandbility} * m \text{ understandbility}) + (w$
 $= \text{ Attractive} * m \text{ Attractive}) + (w \text{ Operability} * m$
 $\text{ Operability})$

$$\begin{aligned} \text{Nilai Suitability} &= (0,61 * 0,61) + (0,47 * 0,28) + (0,34 * 0,11) \\ &= 0.37 + 0.13 + 0.037 \\ &= 0.54 \end{aligned}$$

Dari nilai yang diperoleh adalah 0,54 yang berarti perangkat lunak ini termasuk dalam kategori *medium* atau cukup dalam segi *Usability*.

1. *Efficiency* (efisiensi), perangkat lunak ini memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut. *Efficiency* terdiri atas satu sub karakteristik, yaitu:
 - a. *Time Behaviour* perangkat lunak dapat memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya.
 Nilai faktor *efficiency* adalah 0,53 atau masuk ke dalam kategori *medium* yang berarti perangkat lunak ini cukup efisien.
2. *Maintainability* (pemeliharaan) adalah kemampuan perangkat lunak untuk dimodifikasi. Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional. Sub karakteristik pada aspek *Maintainability* yang digunakan dalam penelitian ini adalah:
 - a. *Changeability*, kemampuan perangkat lunak untuk dimodifikasi dari segi tampilan.
 Nilai yang diperoleh yaitu 0,21 yang berarti perangkat lunak ini termasuk dalam kategori *low* atau kurang dalam segi *maintability*.
3. *Portability* (portabilitas) adalah kemampuan webinar dapat untuk ditransfer dari satu lingkungan ke lingkungan lain. *Portability* memiliki dua sub karakteristik, yaitu:
 - a. *Replaceability*, kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya.
 - b. *Instalability* mampu untuk diinstal dalam lingkungan yang berbeda-beda

Tabel 4. Pembobotan *Portability*

Sub Faktor	Bobot
<i>Replaceability</i>	$W1 = \frac{1+\frac{1}{2}+\frac{1}{3}}{3} = 0.61$
<i>Instability</i>	$W2 = \frac{0+0+\frac{1}{3}}{3} = 0.11$

$$\text{Nilai } Portability = (w \text{ Replaceability} * m \text{ Replaceability}) + (w \text{ Instability} * m \text{ Instability})$$

$$\begin{aligned} \text{Nilai } Portabilit &= (0,82 * 0,61) + (0,49 * 0,11) \\ &= 0.50 + 0.05 \\ &= 0,55 \end{aligned}$$

Dari nilai yang diperoleh adalah 0,55 yang berarti perangkat lunak ini termasuk dalam kategori *medium* atau cukup dari segi *Portability* (portabilitas).

Tabel 5. Perhitungan Nilai Akhir Kualitas Eksternal

Faktor	Bobot
<i>Functionality</i>	$W1 = \frac{1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}}{5} = 0.41$
<i>Usability</i>	$W2 = \frac{0+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}}{5} = 0.21$
<i>Portability</i>	$W3 = \frac{0+0+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}}{5} = 0.16$
<i>Efficiency</i>	$W4 = \frac{0+0+0+\frac{1}{4}+\frac{1}{5}}{5} = 0.09$
<i>Maintability</i>	$W5 = \frac{0+0+0+0+\frac{1}{5}}{5} = 0.04$

$$\text{Nilai Eksternal} = (w \text{ Functionality} * m \text{ Functionality}) + (w \text{ Usability} * m \text{ Usability}) + (w \text{ Portability} * m \text{ Portability}) + (w \text{ efficiency} * m \text{ efficiency}) + (w \text{ maintability} * m \text{ maintability})$$

$$\begin{aligned} \text{Nilai Eksternal} &= (0.48 * 0,41) + (0,53 * 0,21) + (0,55 * 0,16) + (0,53 * 0,09) + (0,21 * 0,04) \\ &= 0,20 + 0,11 + 0,08 + 0,05 + 0,01 \\ &= 0.45 \end{aligned}$$

Dapat disimpulkan bahwa dari nilai akhir kualitas eksternal yaitu 0,45 termasuk ke dalam kategori *medium* atau cukup yang artinya kualitas perangkat *video conference* dari segi eksternal telah mencukupi.

Tanggapan Responden Dan Perhitungan Nilai Metriks Terhadap *Quality In Use* Perangkat Lunak

Tabel 6. Pembobotan *Quality In Use*

Sub Faktor	Bobot
<i>Productivity</i>	$W1 = \frac{1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}}{4} = 0.52$
<i>Safety</i>	$W2 = \frac{0+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}}{4} = 0.27$
<i>Effectiveness</i>	$W3 = \frac{0+0+\frac{1}{3}+\frac{1}{4}}{4} = 0.15$
<i>Satisfaction</i>	$W4 = \frac{0+0+0+\frac{1}{4}}{4} = 0.06$

$$\text{Nilai } Quality In Use = (w \textit{ Productivity} * v \textit{ Productivity}) + (w \textit{ Safety} * v \textit{ Safety}) + (w \textit{ Effectiveness} * v \textit{ Effectiveness}) + (w \textit{ Satisfaction} * v \textit{ Satisfaction})$$

$$\text{Nilai } Quality In Use = (0,72 * 0,52) + (0,91 * 0,27) + (0,54 * 0,15) + (0,92 * 0,06) = 0,37 + 0,25 + 0,08 + 0,06 = 0,76$$

Dapat disimpulkan bahwa dari nilai akhir *Quality In Use* yaitu 0,76 terhadap perangkat lunak termasuk ke dalam kategori *high* atau kualitas baik yang berarti perangkat lunak ini baik dari segi kualitas pemakaian. Hal ini dilihat dan dinilai berdasarkan 4 faktor yang ada pada *Quality In Use* yaitu *Effectiveness* (efektivitas), *productivity* (produktivitas), *Safety* (keamanan) dan *satisfacation* (kepuasan).

KESIMPULAN

Berdasarkan hasil penelitian disimpulkan: 1) faktor *functionality* (fungsionalitas) dengan nilai 0,48 dalam kategori *medium* atau cukup, 2) faktor *Usability* dengan nilai 0,54 dalam kategori *medium* atau cukup, 3) faktor *efficiency* dengan nilai 0,53 atau masuk ke dalam kategori *medium* yang berarti perangkat lunak ini cukup efisien, 4) faktor *maintability* dengan nilai 0,21 yang berarti perangkat lunak ini termasuk dalam kategori *low*, 5) faktor *portability* dengan nilai 0,55 yang berarti perangkat lunak ini termasuk dalam kategori *medium* atau cukup, 6) nilai akhir kualitas eksternal yaitu 0,45 termasuk ke dalam kategori *medium* atau cukup. Berdasarkan hasil penelitian diperoleh nilai paling rendah pada faktor *maintabilit*. Oleh karena itu disarankan kepada pengelola LMS Universitas Negeri Makassar untuk merespon semua keluhan pengguna, memperbaiki dan menyesuaikan LMS dengan perubahan lingkungan dan kebutuhan.



DAFTAR PUSTAKA

- Blokdyk, Gerardus, 2020. *Software Measurement a Complete Guide* , 2020 Edition. Brisbane Australia: Emereo Pty Limited.
- Fenton, Norman & Bieman James, 2014. *Software Metrics*, 3rd Edition, Florida: CRC Press.
- _____, 2015. *Software Metrics, A Rigorous and Practical Approach*, 3rd Edition. Florida: CRC Press.
- Galin, Daniel, 2004. *Software Quality Assurance From Theory to Implementation*, Harlow England: Pearson Education Limited.
- ISO/IEC 25010, *Software and Data Quality*, Online: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>.
- ISO/IEC 25010:2011(en). *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. Online: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>
- Kan, Stephen H, 2003. *Metrics and Models In Software Quality Engineering*. Boston: Pearson Education.
- Kusrini, Dr., M.Kom., 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*, Yogyakarta: Penerbit Andi.
- Laporte, Calude Y & April Alain, 2018. *Software Quality Assurance*, New Jersey USA: John Wiley & Sons, Inc.
- McConnel, Steve, 2004. *Code Complete, A Practical Handbook of Software Construction*, Washington US: Microsoft Press.
- Schulmeyer, G. Gordon , 2008. *Handbook of Software Quality Assurance*, Norwood: Artech House, Inc.