

ADAPTIVE TEST BERDASARKAN REVISED BLOOM'S TAXONOMY MENGGUNAKAN LEARNING VECTOR QUANTIZATION

Eko Subiyantoro¹, Alun Sujjada²

PPPPTK BOE-VEDC Malang¹

Sekolah Tinggi Teknik Atlas Nusantara²

¹ekovedc@gmail.com²alun.sujjada@gmail.com

Abstrak. Paradigma baru dalam pembelajaran menekankan pada pandangan bahwa siswa adalah manusia yang memiliki potensi untuk belajar dan berkembang. Namun alat evaluasi dalam dunia pendidikan umumnya masih menggunakan alat tes yang standar, artinya dalam satu kelas semua siswa mendapat jenis soal yang sama tanpa melihat kemampuan tiap individu. *Revised Bloom's Taxonomy* (RBT) dapat digunakan untuk mengklasifikasi jenis pemikiran yang terlibat dalam proses kognitif siswa. Salah satu cara untuk menyajikan soal sesuai dengan kemampuan kognitif siswa yaitu menggunakan model *adaptive test*.

Learning Vector Quantization (LVQ) merupakan *artificial neural network* terbimbing dengan metode pembelajaran kompetitif yang dapat digunakan untuk mengklasifikasi kemampuan kognitif siswa menjadi tiga kelas kognitif yaitu, *Cognitive Low* (CL), *Cognitive Medium* (CM), dan *Cognitive High* (CH) berdasarkan data *pretest* yang dikerjakan oleh siswa.

Adaptive test mampu merekomendasikan paket soal yang sesuai dengan kelas kompetensi yang diperoleh siswa, kelas CL mendapat rekomendasi soal paket 1, kelas CM mendapat rekomendasi soal paket 2, dan kelas CH mendapat rekomendasi soal paket 3.

Kata Kunci: *rbt, lvq, adaptif test.*

PENDAHULUAN

Di abad ke-21 ini, pendidikan menjadi semakin penting untuk menjamin peserta didik memiliki keterampilan belajar dan berinovasi, keterampilan menggunakan teknologi dan media informasi, serta dapat bekerja dan bertahap dengan menggunakan keterampilan untuk hidup (*life skills*). Paradigma pembelajaran saat ini telah berubah dari pembelajaran tradisional (*transfer of knowledge*) yang berpusat kepada guru menjadi paradigma baru yaitu pembelajaran inovatif (*construction of knowledge*) yang mengedepankan siswa sebagai pusat dari kegiatan pembelajaran. Paradigma baru dalam pembelajaran menekankan pada pandangan bahwa siswa adalah manusia yang memiliki potensi untuk belajar dan berkembang. Siswa harus aktif dalam mencari dan pengembangan pengetahuan yang mereka miliki. Namun penggunaan tes dalam dunia pendidikan umumnya menggunakan alat tes yang standar, artinya dalam satu kelas semua siswa mendapat jenis soal yang sama tanpa melihat kemampuan tiap individu. Maka dari itu pada metode pembelajarannya perlu ditingkatkan, salah satunya adalah dengan memberikan *pretest* sebelum memulai pembelajaran. Setiap individu memiliki kemampuan pikir yang berbeda, dengan melakukan *pretest* maka dapat diketahui kemampuan setiap individu yang kemudian dikelompokkan sesuai hasil yang didapat. Fungsi dari *Revised Bloom's Taxonomy* (RBT) adalah untuk mengklasifikasi jenis pemikiran yang terlibat dalam proses kognitif, yaitu dimulai dari proses berfikir sederhana konkret atau riil hingga pada tingkat berfikir abstrak dan untuk mengklasifikasi jenis kata kerja operasional dalam pembuatan item soal atau tujuan pembelajaran. Pembuatan soal-soal dalam *pretest* didasarkan pada acuan dari RBT dan dikategorikan dalam kelompok C1 (Mengingat), C2 (Memahami), C3 (Menerapkan), C4 (Menganalisis), C5 (Menilai) dan C6 (Mengkreasikan). Dengan demikian diperlukan suatu cara untuk mengklasifikasi tingkatan kognitif masing-masing siswa.

Learning Vector Quantization (LVQ) merupakan metode klasifikasi data yang digunakan dalam penelitian ini. LVQ adalah *Artificial Neural Network* (ANN) yang terbimbing menggunakan metode pembelajaran yang kompetitif dikembangkan oleh (Kohonen et al. 1995), yang digunakan dalam pelatihan terpadu dari lapisan dalam kompetisi ANN. Lapisan kompetitif akan secara otomatis belajar untuk meningkatkan klasifikasi kinerja masukan vektor secara berkala. Ketika beberapa masukan memiliki vektor jarak yang sangat dekat, vektor tersebut akan dikelompokkan dalam beberapa kelas. *Learning Vector Quantization* merupakan salah satu metode jaringan saraf tiruan yang berbasis kompetisi dengan mekanisme *squared euclidian distance* dalam memilih vektor perwakilan pemenang untuk menentukan kategori vektor masukan. Proses pembelajaran LVQ merupakan pembelajaran supervised atau dengan kata lain menggunakan pengarah, dengan tujuan untuk mendapatkan vektor-vektor perwakilan yang akan melakukan kuantisasi terhadap vektor masukan. Dengan metode ini maka pengklasifikasian pola dilakukan dengan setiap unit keluaran mewakili satu kelas tertentu atau satu kategori tertentu.

Adaptive Test adalah suatu metode pengujian atau evaluasi dengan menggunakan teknologi informasi yang bersifat adaptif. Sehingga *adaptive test* dipandang sebagai salah satu solusi dalam membantu siswa dengan mendapatkan soal sesuai dengan kemampuan kognitifnya. *Pretest* disusun berdasarkan *revised bloom's taxonomy* sehingga dapat memenuhi kemampuan kognitif yang dipersyaratkan. Hasil *pretest* kemudian diklasifikasi melalui jaringan *Learning Vector Quantization* dan dikategorikan menjadi tiga kelompok kognitif yaitu *Cognitive Low* (CL), *Cognitive Medium* (CM), dan *Cognitive High* (CH).

Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, permasalahan yang dapat dirumuskan sebagai berikut:

1. Bagaimana melakukan *training* dan pengujian data pretes yang berdasarkan *revised bloom's taxonomy* pada jaringan LVQ?
2. Bagaimana jaringan LVQ dapat menentukan paket soal berdasarkan kelas kompetensi *Cognitive Low* (CL), *Cognitive Medium* (CM), dan *Cognitive High* (CH) ?

Tujuan Penelitian

Berdasarkan latar belakang masalah dan rumusan masalah diatas, maka tujuan dari penelitian adalah:

1. Merancang dan membuat aplikasi *adaptive test* berdasarkan *revised bloom's taxonomy* menggunakan jaringan *Learning Vector Quantization*.
2. Merekomendasikan paket soal yang akan dikerjakan siswa berdasarkan hasil kelas kompetensi yang diperoleh, kelas CL (*Cognitive Low*) mendapat rekomendasi soal paket 1, kelas CM (*Cognitive Medium*) mendapat rekomendasi soal paket 2 dan kelas CH (*Cognitive High*) mendapat rekomendasi soal paket 3.

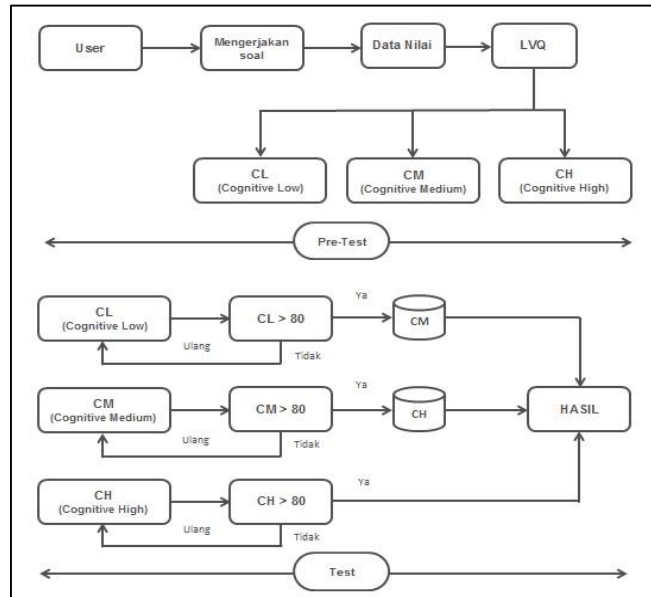
METODE PENELITIAN

Bagian metode penelitian ini menguraikan perancangan aplikasi *adaptive test* dan model *training* dan pengujian data pretest melalui jaringan LVQ.

Rancangan Sistem

Secara umum blok diagram sistem *adaptive test* disajikan dalam Gambar 1. terdapat dua bagian *pretest* dan *test*. Bagian *pretest* merupakan pengujian data melalui jaringan *learning vector quantization* yang menghasilkan tiga klasifikasi kognitif yaitu CL, CM, dan CH. Sedangkan bagian *test* merupakan pengerjaan paket-paket soal sesuai dengan klasifikasi kognitif siswa. Jika hasil yang diperoleh dari mengerjakan paket soal 1 lebih dari 80, maka *user* (siswa) bisa melanjutkan ke tingkat soal CM (*Cognitive Medium*), jika nilai kurang dari 80 bisa mengulangi *test*, sedangkan jika hasil yang diperoleh dari mengerjakan paket soal 2 lebih dari 80, maka *user* (siswa) bisa melanjutkan ke tingkat soal CH (*Cognitive High*), jika nilai kurang dari 80

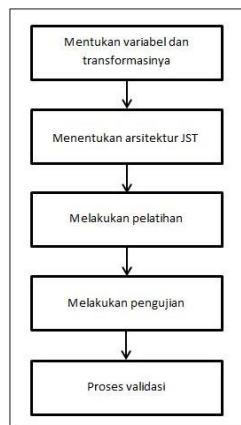
bisa mengulangi *test*. Jika hasil yang diperoleh dari mengerjakan *test* kurang dari 80, maka *user* (siswa) bisa mengulangi *test*



Gambar 1. Blok diagram sistem adaptif Test

Pengujian Sistem

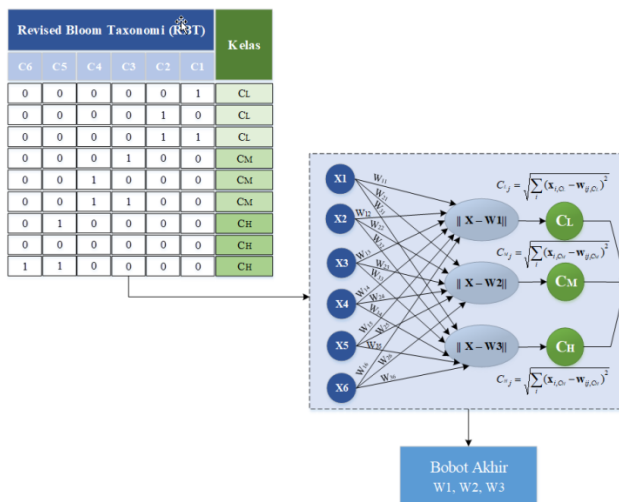
Aplikasi *adaptive test* proses pelatihan jaringan LVQ dengan membuat data training dengan matriks berukuran 15 x 20 yang dikategorikan berdasarkan *revised bloom taxonomy*. Bobot awal dan learning rate ditetapkan sebagai acuan untuk melakukan proses penghitungan dalam menentukan jaringan LVQ selengkapya tersaji dalam Gambar 2.



Gambar 2. Model pengujian jaringan Learning Vector Quantization

Arsitektur Jaringan LVQ

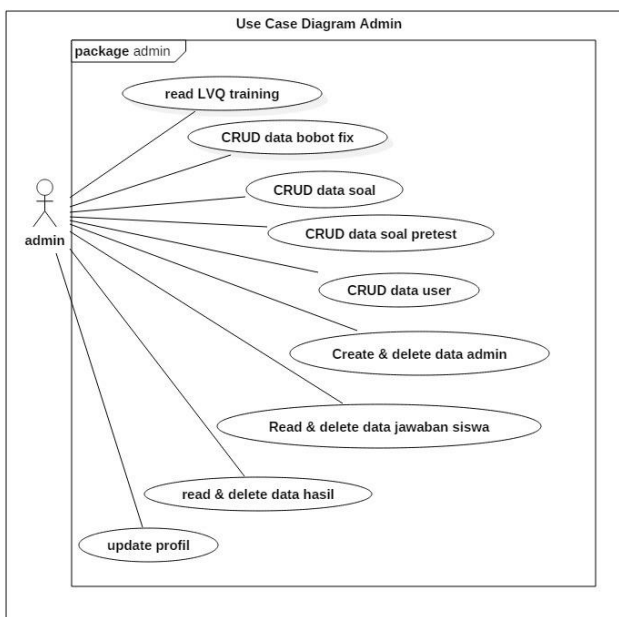
LVQ digunakan untuk mengklasifikasikan enam data vektor masukan yaitu mengingat(C1),memahami(C2),menerapkan(C3),menganalisis(C4),mengevaluasi(C5), dan mencipta(C6). Variabel dari vektor input dapat dituliskan seperti berikut, yaitu $X = \{ X_1, X_2, X_3, X_4, X_5, X_6 \}$ dengan neuron keluaran $Y_1, Y_2, \text{ dan } Y_3$, serta tiga vektor bobot yaitu $W_{1j} = \{ W_{11}, W_{12}, W_{13}, W_{14}, W_{15}, W_{16} \}$, $W_{2j} = \{ W_{21}, W_{22}, W_{23}, W_{24}, W_{25}, W_{26} \}$, dan $W_{3j} = \{ W_{31}, W_{32}, W_{33}, W_{34}, W_{35}, W_{36} \}$. Maka akan mempunyai ilustrasi dalam Gambar 3.



Gambar 3. Arsitektur jaringan LVQ

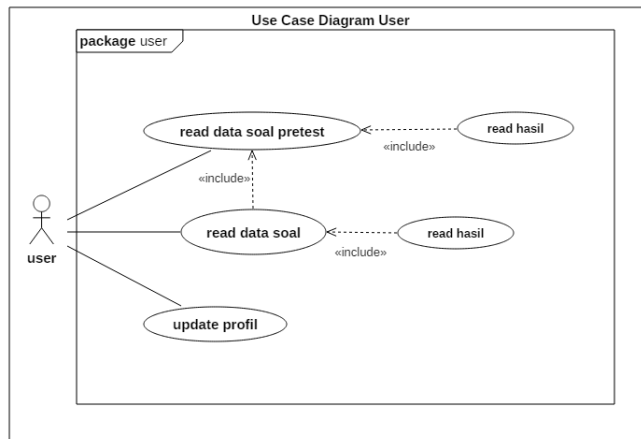
Use Case Diagram Sistem Adaptive Test

Pada *use case* diagram admin disajikan dalam Gambar 4. yang dapat diakses oleh admin yaitu LVQ training, data bobot fix, data soal pretest, data soal, data user, data admin, data jawaban siswa dan data hasil.



Gambar 4. Use case diagram admin

Pada *use case* diagram user disajikan dalam Gambar 5, yang dapat diakses oleh user yaitu melihat data soal pretest, jika ingin melihat data soal user harus mengerjakan ujian pretest terlebih dahulu.



Gambar5. Use case diagram user

HASIL DAN PEMBAHASAN

Bagian pembahasan terdapat beberapa tahapan, yaitu: *user* harus *login* sebelum melakukan ujian *pretest*, jika belum punya akun *user* harus mendaftar pada tautan yang menuju *form* pendaftaran, jika *user* sudah punya akun dan terdaftar *user* akan dialihkan ke halaman *dashboard* dan bisa melakukan ujian *pretest* dan bisa langsung melihat hasilnya.

Analisa Pengujian Jaringan LVQ

Pada bagian ini akan dijelaskan tahapan yang dilakukan dalam pengujian pada halaman LVQ *Training* bagian admin yang bertujuan untuk mendapatkan bobot *fix*:

- a. Alur perhitungan LVQ *Training*.



Gambar6. Alur *training* data

Data berukuran 15x20 maksudnya adalah siswa berjumlah 15 dan soal berjumlah 20.

- b. Input data *training* berukuran 15x20 dengan target yang telah ditentukan seperti pada Gambar 7 berikut.

No	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	Target
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
3	1	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1
4	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1
5	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1
6	1	1	1	0	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	2
7	1	1	1	0	1	0	1	1	1	1	1	1	0	0	1	0	1	0	1	0	2
8	0	1	0	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	0	2
9	1	1	0	1	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	2
10	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	2
11	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	0	1	1	0	3
12	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	1	1	3
13	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	3
14	1	1	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	3
15	0	1	1	0	1	1	1	1	0	0	1	1	1	0	0	1	1	1	1	1	3

Gambar7. Data *Training*

- c. Bobot awal yang telah ditentukan seperti pada Gambar 8. berikut.

No	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	Target	
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-CL
2	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	2-CM
3	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1	0	1	0	3-CH

Gambar8. Bobot awal

- d. Kemudian data-data diatas dihitung menggunakan metode *Learning Vector Quantization* yang telah dibuat dalam bentuk aplikasi *web* dengan memasukkan jumlah *epoch* dan *learning rate (alpha)* sesuai kebutuhan aplikasi. Dari hasil perhitungan yang digunakan untuk aplikasi Adaptive Test ini dengan *input* jumlah *epoch* = 1 dan *learning rate (alpha)* = 0,05 dihasilkan bobot akhir seperti pada gambar 9 tiap baris urut dari atas mewakili W1, W2 dan W3.

0.95	0.90	1.00	0.86	0.95	0.95	0.82	0.04	0.05	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.00	0.00
0.90	1.00	0.95	0.91	1.00	0.91	1.00	0.96	1.00	1.00	0.09	1.00	0.95	0.09	0.00	0.05	0.05	0.05	0.05	0.05	0.00
0.95	1.00	1.00	0.95	0.95	1.00	0.91	1.00	0.09	0.86	1.00	1.00	0.95	0.05	0.05	0.95	1.00	1.00	0.10	0.95	

Gambar9. Bobot fix

- e. Tahap selanjutnya setelah bobot *fix* didapatkan maka akan dilakukan pengujian dengan data *training* baru menggunakan metode LVQ. Berikut adalah contoh data *training* baru yang digunakan.

No	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	Target
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
3	1	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1
4	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1
5	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1
6	1	1	1	0	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	2
7	1	1	1	0	1	0	1	1	1	1	1	1	0	0	0	1	0	1	0	0	2
8	0	1	0	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	0	2
9	1	1	0	1	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	2
10	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	2
11	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	0	1	1	0	3
12	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	1	1	3
13	1	1	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	3
14	1	1	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	3
15	0	1	1	0	1	1	1	1	0	0	1	1	1	0	0	1	1	1	1	1	3

Gambar10 Data pengujian

- f. Dari hasil pengujian data *training* baru dengan jumlah data 15 dan bobot akhir didapatkan hasil seperti pada gambar 4.26 dengan persentase keakuratan 100%.

Data Siswa 1 T:(1) masuk dalam Kelas 1
Data Siswa 2 T:(1) masuk dalam Kelas 1
Data Siswa 3 T:(1) masuk dalam Kelas 1
Data Siswa 4 T:(1) masuk dalam Kelas 1
Data Siswa 5 T:(1) masuk dalam Kelas 1
Data Siswa 6 T:(2) masuk dalam Kelas 2
Data Siswa 7 T:(2) masuk dalam Kelas 2
Data Siswa 8 T:(2) masuk dalam Kelas 2
Data Siswa 9 T:(2) masuk dalam Kelas 2
Data Siswa 10 T:(2) masuk dalam Kelas 2
Data Siswa 11 T:(3) masuk dalam Kelas 3
Data Siswa 12 T:(3) masuk dalam Kelas 3
Data Siswa 13 T:(3) masuk dalam Kelas 3
Data Siswa 14 T:(3) masuk dalam Kelas 3
Data Siswa 15 T:(3) masuk dalam Kelas 3

Gambar11 Hasil pengujian LVQ

Hasil Pengujian Adaptive Test

Halaman LVQ *Training*, menampilkan data *training* dan bobot awal berupa tabel dan *form* perhitungan LVQ dengan memasukkan jumlah *epoch*, *learning rate* (*alpha*) dan konstanta beta (opsional).

The screenshot shows the 'LVQ Training' page. On the left is a sidebar menu. The main area has a title 'LVQ Training' and a user profile 'admin ADMINISTRATOR'. Below the title is a warning box: 'Input data dari hasil (berupa bobot fix) dari proses training tidak bersifat dinamis, jika ingin mengubah jumlah data atau bobot awal dilakukan secara manual dengan mengubah basis program (proses ini dilakukan oleh programmer)'. Below this is a 'Data Training' table with 7 rows and 15 columns (X1 to X15). To the right is a 'Form perhitungan LVQ' with input fields for 'Jumlah EPOCH', 'Learning Rate (alpha)', and 'Konstanta BETA', and a 'HITUNG' button.

Gambar12. Halaman LVQ training

Halaman *Bobot Fix*, menampilkan data bobot yang digunakan untuk aplikasi ini dalam bentuk tabel. Data bobot *fix* diperoleh dari perhitungan LVQ *Training*. Terdapat juga tombol-tombol untuk menunjang admin untuk memanajemen data berupa tombol tambah, ubah dan hapus.

The screenshot shows the 'Robot Fix' page. The sidebar menu is on the left. The main area has a title 'Robot Fix' and a user profile 'admin ADMINISTRATOR'. Below the title is a 'Tabel Bobot Fix' with a search bar and a '+ TAMBAH BOBOT' button. The table has 2 rows and 3 columns: ID, Bobot, and Aksi. Each row has 'UBAH' and 'HAPUS' buttons. At the bottom, there are 'Previous' and 'Next' navigation buttons.

Gambar13. Halaman Bobot Fix



Gambar14. Tampilan halaman output adaptif test

PENUTUP

Dari hasil pembuatan aplikasi Adaptive Test Berdasarkan Revised Bloom Taxonomy menggunakan jaringan *Learning Vector Quantization* dapat diambil beberapa kesimpulan dan saran sebagai berikut.

Kesimpulan

1. *Training* LVQ menggunakan data dengan matriks ukuran 15x20 berhasil mendapat bobot *fix* dan keakuratan 100%. Keakuratan LVQ diukur dengan jumlah benar dikurangi jumlah salah dibagi jumlah data.
2. Hasil *pretest* akan merekomendasikan paket soal yang didapat berdasarkan hasil kelas kompetensi yang diperoleh, kelas CL (*Cognitive Low*) mendapat rekomendasi soal paket 1, kelas CM (*Cognitive Medium*) mendapat rekomendasi soal paket 2 dan kelas CH (*Cognitive High*) mendapat rekomendasi soal paket 3.

Saran

1. Diharapkan jika ingin menambah jumlah data soal tidak lagi dilakukan secara manual dengan mengganti kode program.
2. Diharapkan jika ingin mengubah data bobot *fix* tidak lagi dilakukan secara manual dengan mengganti kode program.
3. Mengembangkan metode lebih lanjut agar keakuratan hasil yang diperoleh menjadi lebih besar.

DAFTAR RUJUKAN

- Anderson, L. W., Kratchwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., et al., 2001. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Longman.
- Andi, 2010. *PanduanPraktisMenguasaiPemrograman Web dengan JavaScript*. Semarang: Wahana Komputer.
- Ardiansyah, Doni, 2017. *Aplikasi Penerimaan Siswa Baru Online Dengan PHP, PDO, dan Bootstrap*. Cirebon: ASFA Solution.
- Budianita E. & Widodo Prijodiprodjo, 2013. *Penerapan Learning Vector Quantization untuk Kalifikasi Status Gizi Anak*, Vol. 7, No. 2, 155-156.
- Conklin, J., 2005. *Book Reviews: A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. *Educational Horizons*, 83, 154-159.
- Faridl, Miftah, 2015. *Fitur Dahsyat Sublime Text 3*, <http://lug.stikom.edu/wp-content/media/Fitur-Dahsyat-Sublime-Text-3.pdf> (diakses pada tanggal 18 Mei 2018)
- Jubilee Enterprise, 2017. *Otodidak Desain dan Pemrograman Website*. Jakarta: Elex Media Komputindo.
- Kusumadewi, Sri, 2003. *Artificial Itelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Krathwohl, D. R., 2002. *A Revision of Bloom's Taxonomy: An Overview*. *Theory into Practice*, 41(4).
- Kohonen T., Hynninen J., Kangas J., Laaksonen J. and Torkkola K. (1995), *LVQ_PAK The Learning Vector Quantization Program Package Version 3.1 (April 7, 1995)*, The LVQ Programming Team of the Helsinki University of Technology Laboratory of Computer and Information Science, Rakentajanaukio 2 C, SF-02150 Espoo FINLAND, Finland
- Masters. N.G. & Keeves. P.J., 1999. *Advances in measurement in educational research and assesment*: New York: Pergamon Press.
- Muslihudin, Muhamad dan Oktafianto, 2016. *Analisis dan Perancangan Sistem. Informasi Menggunakan Model Terstruktur dan UML*. Yogyakarta: Andi Offset.
- Nurjannah F., *Notasi UML*, http://fitri_nurjannah.staff.gunadarma.ac.id/Downloads/folder/0.1 (diakses pada tanggal 18 Mei 2018).
- Weiss. D.J., 2004. *Computerized Adaptive Testing for Effective and Efficient Measurement in Counseling and Education*. *Measurement and Evaluation in Counseling and Development*, Vol. VII, No: 37: 70-75.
- Yuhefizar, Mooduto & Rahmat H., 2009. *Cara Mudah Membangun Website Interaktif Menggunakan Content Management System Joomla*. Jakarta: Elex Media Komputindo.