Analizy OSW

OŚRODEK STUDIÓW WSCHODNICH 27.11.2026 www.osw.waw.pl

Turcja po COP31: kapitał i technologie mimo fiaska negocjacji

Adam Michalski

W dniach 9–22 listopada w Antalyi odbyła się konferencja klimatyczna ONZ (COP31), która po raz pierwszy od 17 lat zakończyła się bez przyjęcia deklaracji końcowej. Przyczyną impasu był spór o zapisy dotyczące odchodzenia od paliw kopalnych; ich podtrzymania domagały się UE, Wielka Brytania i część państw Globalnego Południa, a sprzeciwiali się temu eksporterzy węglowodorów (m.in. Arabia Saudyjska i Rosja). Ostateczny projekt porozumienia, wypracowany przez sprawującą prezydencję COP31 Turcję i kierującą negocjacjami Australię, pominął tę kwestię, dlatego m.in. UE uznała deklarację za zbyt słabą i ją odrzuciła.

Po szczycie przewodniczący COP31 Murat Kurum skrytykował postawę państw europejskich, podkreślając korzyści gospodarcze konferencji dla Ankary. Należą do nich deklaracje inwestycyjne i umowy finansowe o łącznej wartości ponad 26 mld dolarów, obejmujące m.in. współpracę z Chinami w zakresie energii odnawialnej (OZE) oraz finansowanie transformacji energetycznej ze strony banków rozwoju (zob. tabela). Turcja zawarła także porozumienia z Arabią Saudyjską dotyczące OZE oraz z kanadyjską spółką Candu Energy w sprawie rozwoju energetyki jądrowej.

Fiasko rozmów klimatycznych wynikało zarówno z trudności w pogodzeniu sprzecznych interesów uczestników szczytu, jak i ze sposobu prowadzenia negocjacji przez Ankarę i Canberrę. Sposób ten uwypuklił potencjalną tendencję Turcji do faworyzowania postulatów państw, z którymi blisko współpracuje w dziedzinie energetyki i bezpieczeństwa. Mimo niepowodzenia negocjacji Ankara uzyskała deklaracje finansowania inwestycji oraz transferu zielonych technologii do krajowego przemysłu.

Komentarz

- **Organizacja konferencji COP31 miała ugruntować wizerunek Turcji jako państwa zdolnego do godzenia sprzecznych interesów i osiągania kompromisów na forum międzynarodowym.** Po sukcesie lipcowego szczytu NATO w Ankarze COP31 miała potwierdzić skuteczność tureckiej dyplomacji również w negocjacjach klimatycznych. Turcja dążyła do odegrania roli pośrednika między państwami postulującymi przyspieszenie transformacji energetycznej a eksporterami węglowodorów, zainteresowanymi utrzymaniem znaczenia paliw kopalnych w światowej gospodarce. Jednocześnie COP31 miała służyć nie tylko wypracowaniu wspólnych zobowiązań klimatycznych, lecz także mobilizacji kapitału i upowszechnianiu technologii niezbędnych do ich realizacji, na czym miała skorzystać również Turcja.

- **Dyplomatyczne ambicje Ankary zderzyły się ze sprzecznymi interesami gospodarczymi uczestników szczytu.** Państwa eksportujące ropę i gaz, którym zależy na długoterminowym utrzymaniu dochodów z ich sprzedaży, dążyły do zachowania aktualnego poziomu wydobycia tych surowców, co kolidowało z postulatami UE i państw szczególnie narażonych na skutki zmian klimatu. Turcja próbowała pogodzić te stanowiska, proponując uwzględnienie w deklaracji celu zwiększenia udziału energii elektrycznej w światowym końcowym zużyciu energii do 35% do 2035 r. Oznaczałoby to zastępowanie paliw kopalnych energią elektryczną (np. poprzez rozwój elektromobilności), ale bez zobowiązywania państw do ograniczania wykorzystania tych surowców do produkcji prądu. Formuła ta, choć nie godziła w interesy eksporterów węglowodorów, okazała się niewystarczająca dla krajów domagających się odchodzenia od paliw kopalnych.
- **Sposób prowadzenia negocjacji w końcowej fazie szczytu dodatkowo osłabił zaufanie części delegacji do Ankary jako bezstronnego pośrednika.** Według doniesień medialnych delegacja Arabii Saudyjskiej miała regularny dostęp do przedstawicieli tureckiej prezydencji, podczas gdy innym trudno było umówić się na spotkanie. Ankara zaprzeczała tym zarzutom. Wątpliwości te pogłębił ostateczny projekt porozumienia. Zgodnie z postulatem Rijadu organizatorzy COP31 utrzymali w deklaracji zapis o szerszym wykorzystaniu technologii wychwytywania CO_2, pozwalających ograniczać emisje bez rezygnacji ze spalania paliw kopalnych. Usunięcia tego zapisu domagało się 126 państw w obawie, że posłuży on za pretekst do opóźniania odchodzenia od tych surowców. Ustępstwa wobec Rijadu i stolic o podobnym stanowisku poczyniono zatem kosztem poparcia ze strony innych delegacji, których zgoda była niezbędna do przyjęcia porozumienia.
- **Mimo fiaska negocjacji Ankara wykorzystała COP31 do pozyskania deklaracji wsparcia kapitałowego i technologicznego dla transformacji energetycznej kraju.** Proces ten ma służyć ograniczeniu kosztownego importu surowców energetycznych oraz utrzymaniu konkurencyjności tureckiego przemysłu na rynku UE, gdzie rosną koszty emisji i zaostrzają się wymogi klimatyczne. Potrzebne jest jednak zagraniczne finansowanie rozbudowy OZE, magazynów energii i infrastruktury przesyłowej. Istotny jest również przewidziany w części porozumień transfer technologii wspierający rozwój lokalnych fabryk (m.in. wytwarzających komponenty do turbin wiatrowych we współpracy z Chinami), który może wzmocnić pozycję Turcji jako zaplecza produkcyjnego sektora zielonych technologii.

' + ' □'; root.append(pasek); const btn = pasek.querySelector(".osw-pobierz-btn"), panel = pasek.querySelector(".osw-pobierz-panel"), inp = panel.querySelector("input"), bZap = panel.querySelector(".osw-zapisz"), bZam = panel.querySelector(".osw-zamknij"); btn.title = T.btn; btn.setAttribute("aria-label", T.btn); btn.setAttribute("aria-expanded", "false"); panel.querySelector("span").textContent = T.szer; bZap.textContent = T.zapisz; bZam.title = T.zamknij; bZam.setAttribute("aria-label", T.zamknij); inp.value = root.getAttribute("data-png-szer") || 1200; // Położenie przycisku: prawy górny róg tabeli, jeśli nie zasłoni napisów w wierszach nagłówka; // inaczej pod prawym dolnym rogiem – opływany przez źródło lub przypis (bez kolizji z ich treścią) albo w osobnym wierszu. const przecina = (a, b) => a.left < b.right && a.right > b.left && a.top <

b.bottom && a.bottom > b.top; function rogWolny() { if (root.getAttribute("data-ikona") === "pod") return null; const pw = root.querySelector(".osw-przewijak"), tab = root.querySelector("table.report-table"); if (!pw || !tab || !pw.offsetWidth || !tab.offsetWidth) return null; // karty – tabela ukryta if (root.classList.contains("osw-mob") && root.getAttribute("data-mobile") === "przewijanie") return null; // przewijana tabela – róg się przesuwa const odst = +root.getAttribute("data-rog") || 4, bw = btn.offsetWidth || 30, bh = btn.offsetHeight || 30, P = pw.getBoundingClientRect(); const r = { right: Math.min(P.right, tab.getBoundingClientRect().right) - odst, top: P.top + odst }; r.left = r.right - bw; r.bottom = r.top + bh; const z = { left: r.left - 4, right: r.right + 4, top: r.top - 3, bottom: r.bottom + 3 }; const wiersze = []; for (const tr of tab.rows) { if (!/\brole-(h1|h2a|h2b)\b/.test(tr.className)) break; wiersze.push(tr); } // wiersze nagłówka na górze tabeli if (!wiersze.length || wiersze[wiersze.length - 1].getBoundingClientRect().bottom < r.bottom) return null; // przycisk wyszedłby poza nagłówek const rg = document.createRange(); for (const tr of wiersze) { const w = document.createTreeWalker(tr, NodeFilter.SHOW_TEXT); while (w.nextNode()) { if (!w.currentNode.nodeValue.trim()) continue; rg.selectNodeContents(w.currentNode); for (const x of rg.getClientRects()) if (x.width && przecina(x, z)) return null; } } return pw; } let pozycja = ""; const ustawPoz = () => { if (!panel.hidden) return; // nie przestawiaj otwartego panelu const pw = rogWolny(); if (pw) { if (pozycja !== "rog" || pasek.parentNode !== pw) { pw.append(pasek); pasek.setAttribute("data-poz", "rog"); pozycja = "rog"; } return; } // pod tabelą: pierwszy widoczny przypis albo źródło – przycisk opływany z prawej let cel = null; for (const el of root.querySelectorAll(":scope > .table-notes .fn-note, :scope > .table-source")) if (el.offsetHeight) { cel = el; break; } if (cel) { if (pasek.parentNode !== cel) { cel.insertBefore(pasek, cel.firstChild); } pasek.setAttribute("data-poz", "oplyw"); pozycja = "oplyw"; } else { if (pasek.parentNode !== root) root.append(pasek); pasek.setAttribute("data-poz", "pod"); pozycja = "pod"; } }; ustawPoz(); if (window.ResizeObserver) new ResizeObserver(() => requestAnimationFrame(ustawPoz)).observe(root); else window.addEventListener("resize", ustawPoz); if (document.fonts && document.fonts.ready) document.fonts.ready.then(ustawPoz); const otworzPanel = on => { panel.hidden = !on; if (!on) setTimeout(ustawPoz, 0); btn.setAttribute("aria-expanded", on ? "true" : "false"); btn.classList.toggle("akt", on); if (on) { inp.focus(); inp.select(); } }; btn.addEventListener("click", e => { e.stopPropagation(); otworzPanel(panel.hidden); }); bZam.addEventListener("click", e => { e.stopPropagation(); otworzPanel(false); btn.focus(); }); pasek.addEventListener("click", e => e.stopPropagation()); pasek.addEventListener("pointerover", e => e.stopPropagation()); panel.addEventListener("keydown", e => { if (e.key === "Enter") { e.preventDefault(); zapisz(); } else if (e.key === "Escape") { otworzPanel(false); btn.focus(); } }); bZap.addEventListener("click", zapisz); let fonty = null; async function css64(url, filtr) { const css = await (await fetch(url)).text(); let wynik = "", m; const re = /(?:\/*\s*([\w-]+)\s**\/\s*)?@font-face\s*\{([^}]*)\}/g; while ((m = re.exec(css))) { if (filtr && m[1] && m[1] !== "latin" && m[1] !== "latin-ext") continue; const src = (/url\(\s*['"]?([^'")]+)['"]?\s*\)/.exec(m[2]) || [])[1]; if (!src) continue; const b = await (await fetch(new URL(src, new URL(url, location.href)).href)).blob(); const d = await new Promise((ok, zle) => { const r = new FileReader(); r.onload = () => ok(r.result); r.onerror = zle; r.readAsDataURL(b); }); wynik += "@font-face{" + m[2].replace(/url\([^)]+\)/, "url(" + d + ")") + "}\n"; } return wynik; } async function stylFontow() { if (fonty !== null) return fonty; for (const [u, f] of [[root.getAttribute("data-fonty"), true], [root.getAttribute("data-fonty-lok"), false]]) { if (!u) continue; try { fonty = await css64(u, f); if (fonty) return fonty; } catch (e) {} } return (fonty = ""); } async function obraz(szer) { const { svg: c, w, h, tlo } = przygotuj(); let wys = Math.max(1, Math.round(szer * h / w)); const k = Math.min(1, Math.sqrt(MAX_POW / (szer * wys)),

MAX_BOK / wys, MAX_BOK / szer); if (k < 1) { szer = Math.floor(szer * k); wys = Math.max(1, Math.round(szer * h / w)); } c.setAttribute("width", szer); c.setAttribute("height", wys); if (!c.getAttribute("xmlns")) c.setAttribute("xmlns", "http://www.w3.org/2000/svg"); const st = await stylFontow(); if (st) { const e = document.createElementNS("http://www.w3.org/2000/svg", "style"); e.textContent = st; c.insertBefore(e, c.firstChild); } const img = new Image(); await new Promise((ok, zle) => { img.onload = ok; img.onerror = zle; img.src = "data:image/svg+xml;charset=utf-8," + encodeURIComponent(new XMLSerializer().serializeToString(c)); }); if (img.decode) try { await img.decode(); } catch (e) {} const cv = document.createElement("canvas"); cv.width = szer; cv.height = wys; const ctx = cv.getContext("2d"); if (tlo) { ctx.fillStyle = tlo; ctx.fillRect(0, 0, szer, wys); } ctx.drawImage(img, 0, 0, szer, wys); const blob = await new Promise((ok, zle) => cv.toBlob(b => b ? ok(b) : zle(new Error("toBlob")), "image/png")); return { blob, szer, obciete: k < 1 }; } async function zapisz() { let szer = Math.round(Math.min(10000, Math.max(100, +inp.value || 1200))); inp.value = szer; const nazwa = s => "OSW_" + (root.getAttribute("data-png") || "grafika") + "_" + s + "px.png"; let uchwyt = null; if (window.showSaveFilePicker) { try { uchwyt = await window.showSaveFilePicker({ suggestedName: nazwa(szer), types: [{ description: "PNG", accept: { "image/png": [".png"] } }] }); } catch (e) { if (e && e.name === "AbortError") return; uchwyt = null; } } bZap.disabled = true; bZap.textContent = T.czekaj; try { const w = await obraz(szer); if (w.obciete) { inp.value = w.szer; } if (uchwyt) { const z = await uchwyt.createWritable(); await z.write(w.blob); await z.close(); } else { const a = document.createElement("a"); a.href = URL.createObjectURL(w.blob); a.download = nazwa(w.szer); document.body.append(a); a.click(); a.remove(); setTimeout(() => URL.revokeObjectURL(a.href), 4000); } if (w.obciete) alert(T.obciete + w.szer + " px"); otworzPanel(false); btn.focus(); } catch (e) { alert(T.blad); } finally { bZap.disabled = false; bZap.textContent = T.zapisz; } } })(root,(function przygotujTabeli(root, svgTekst) { return function () { const d = document.createElement("div"); d.innerHTML = svgTekst; const c = d.querySelector("svg"); for (const a of ["class", "style", "aria-hidden", "focusable", "hidden"]) c.removeAttribute(a); const vb = c.viewBox.baseVal; return { svg: c, w: vb.width, h: vb.height, tlo: "#ffffff" }; }; })(root,D.svg));})();